

Techniques for Enhancing Digital Images

Author:

William E. Harrison III

Supervisor:

Dr. Weiping Lu

*A thesis submitted in fulfillment of the requirements for the degree
of Master of Philosophy.*

in

Heriot Watt University

School of Engineering and Physical Sciences

December 2016

"The copyright in this thesis is owned by the author. Any quotation from the thesis or use of any of the information contained in it must acknowledge this thesis as the source of the quotation or information."

Declaration

Declaration of Authorship

I, William E Harrison, III, declare that this thesis titled, 'Techniques for Enhancing Digital Images', and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:

Date:

ABSTRACT

The images obtain from either research studies or optical instruments are often corrupted with noise. Image denoising involves the manipulation of image data to produce a visually high quality image. This thesis reviews the existing denoising algorithms and the filtering approaches available for enhancing images and/or data transmission.

Spatial-domain and Transform-domain digital image filtering algorithms have been used in the past to suppress different noise models. The different noise models can be either additive or multiplicative. Selection of the denoising algorithm is application dependent. It is necessary to have knowledge about the noise present in the image so as to select the appropriated denoising algorithm. Noise models may include Gaussian noise, Salt and Pepper noise, Speckle noise and Brownian noise. The Wavelet Transform is similar to the Fourier transform with a completely different merit function. The main difference between Wavelet transform and Fourier transform is that, in the Wavelet Transform, Wavelets are localized in both time and frequency. In the standard Fourier Transform, Wavelets are only localized in frequency. Wavelet analysis consists of breaking up the signal into shifted and scales versions of the original (or mother) Wavelet. The Wiener Filter (mean squared estimation error) finds implementations as a LMS filter (least mean squares), RLS filter (recursive least squares), or Kalman filter.

Quantitative measure (metrics) of the comparison of the denoising algorithms is provided by calculating the Peak Signal to Noise Ratio (PSNR), the Mean Square Error (MSE) value and the Mean Absolute Error (MAE) evaluation factors. A combination of metrics including the PSNR, MSE, and MAE are often required to clearly assess the model performance.

Acknowledgement

I wish to express my deepest gratitude to Dr. Weiping Lu, Dr. Euan Brown, and Dr. Roy Duncan for their assistance, guidance, encouragement, and friendship throughout my graduate career at Heriot Watt University. I would also wish to thank Dr. Euan Brown for his encouragement when I took a temporary leave due to unexpected Medical problems.

Contents

	Page No.
Declaration of Authorship	i
Abstract	ii
Acknowledgements	iii
Contents	iv
List of Tables	viii
List of Figures	ix
List of Images	xii
 Chapter	
1. Introduction.....	1
1.1 Fundamentals of Digital Image.....	1
1.2 Noise in Digital Image.....	4
1.3 Gaussian Noise.....	5
1.4 Salt and Pepper Noise.....	6
1.5 Speckle Noise.....	7
1.6 Brownian Noise Speckle.....	8
1.7 Quantifying Noise.....	9
1.7.1 Mathematics	9
1.8 Objectives.....	10
1.9 Conclusion.....	12
 2. Image Processing Basics.....	 13

Table of Contents

2.1	Digital Image.....	13
2.2	Processing Methods.....	18
2.3	Spatial Domain Methods.....	19
2.3.1	Mean Filter.....	19
2.3.2	Median Filter.....	21
2.3.3	Non-local Means Filter.....	23
2.4	Frequency Domain Filter.....	26
2.4.1	Butterworth Filter.....	26
2.4.2	Gaussian Filter-Convolution.....	27
2.4.3	Wiener Filter	29
2.4.3.1	Point Spread Function-Wiener Filter.....	31
2.4.3.2	Application of Wiener Filter.....	32
2.5	Algorithm.....	35
3.	Wavelets.....	36
3.1	Introduction.....	36
3.2	Wavelet Transform.....	37
3.2.1	Preview	37
3.2.2	Simplest Example of Image Transform- Walsh Transform.....	40
3.2.3	The Haar Wavelet.....	42
3.3	Image Denoising.....	44
3.4	Discrete Wavelet Transform.....	45
3.5	Wavelet Thresholding.....	48
3.5.1	Visushrink	50
3.5.2	SureShrink.....	51
3.5.3	Bayes Shrink.....	52
3.6	Wavelet Filters in MATLAB.....	54
3.7	VisuShrink Example.....	56
3.8	SureShrink Example.....	58
3.9	BayesShrink Example.....	59
3.10	Summary.....	63
4.	Image Processing Applications in Science and Medicine...	64
4.1	Astronomy.....	64
4.2	Medicine.....	65

Table of Contents

4.2.1	Tomographic Imaging.....	66
4.2.1.1	Transmissive Mode.....	67
4.2.1.2	Emissive Mode.....	67
4.2.1.3	Reflective Mode.....	68
4.3	Experimental Results.....	74
4.3.1	Performance Evaluation – Denoising Technique	74
4.3.2	Table of Results Comparison.....	80
4.4	Generalization into other Scenarios	
	3D MRI Imaging	82
4.4.1	Create Head Phantom – Two Dimensions	83
4.4.2	Extension to creating 3D MRI Images	84
5.	Conclusions.....	90
6.	Scope for Future Work.....	91
6.1	Fuzzy Methods.....	91
6.2	Vision 3-D Map.....	92
6.3	Medical Image Computing.....	92
7.	References.....	95
	APPENDIX A: MATLAB FUNCTIONS.....	100
	APPENDIX A: Latest Versions MATLAB.....	102
	APPENDIX A: MATLAB Convolution Denoised.....	103
	APPENDIX A: MATLAB Airy Point Spread Function.....	104
	APPENDIX A: MATLAB Image Denoised.	105
	APPENDIX B: Medical Glossary.....	107

Table of Contents

APPENDIX C: C Code..... 109

List of Tables

Table 2.1 List of Important Denoising Techniques in the Spatial and Frequency Domain	18
Table 2.2 Metrics for Denoised Lena Image with Median Filter	22
Table 3.1 Performance Evaluation Metrics Wavelet db4	60
Table 3.2 Performance Evaluation Metrics Wavelet sym4	61
Table 3.3 Performance Evaluation Metrics Wavelet bior 6.8	62
Table 4.1 Quality Analyses (MRI Image) - GLOBAL THRESHOLDING ...	80
Table 4.2 Quality Analyses (MRI Image) - LEVEL DEPENDENT THRESHOLDING	81

List of Figures

Figure 1.1 Gaussian distribution	5
Figure 1.2 PDF for salt and pepper noise	6
Figure 1.3 Gamma Distribution	7
Figure 1.4: Brownian noise distribution	8
Figure 2.1 The discrete pixel numbering convention	14
Figure 2.2 Using different colormaps to display the same binary image ...	14
Figure 2.3 A greyscale image	15
Figure 2.4 Default Color Maps available in Matlab	16
Figure 2.5 The Shift-Multiply-Sum Principle	20
Figure 2.6 Illustrates an example calculation	21
Figure 3.1 Haar wavelets of various scales and translations	43
Figure 3.2 Block Diagram of Image denoising using Wavelet transform ...	44
Figure 3.3 First Level Decomposition	46
Figure 3.4 Second Level Subbands	46
Figure 3.5 Third Level Subbands	46
Figure 3.6 Hard thresholding	49
Figure 3.7 Soft thresholding	50
Figure 3.8 Wavelet Families	54

Figure 3.9 Wavelet Families	54
Figure 3.10 Wavelet Families	55
Figure 4.1 Full Moon	65
Figure 4.2 Full Moon	65
Figure 4.3 Full Moon	65
Figure 4.4 Ax T1 SE slice #1	69
Figure 4.5 Ax T1 SE slice #6	69
Figure 4.6 Ax T1 SE slice #12	69
Figure 4.7 Ax T1 SE slice #14	69
Figure 4.8 MRA Carotid/3-Plane Fiesta S7	70
Figure 4.9 MRA Carotid/3-Plane Fiesta S8	70
Figure 4.10 MRA Carotid/3-Plane Fiesta S9	70
Figure 4.11 MRA Carotid/3-Plane Fiesta S10	70
Figure 4.12 MRA Carotid/3-Plane Fiesta S11	71
Figure 4.13 MRA Carotid/3-Plane Fiesta S12	71
Figure 4.14 MRA Carotid/3-Plane Fiesta S15	71
Figure 4.15 MRA Carotid/3-Plane Fiesta S18	71
Figure 4.16 MRA Carotid/3-Plane Fiesta	72
Figure 4.17 CT Head WO Contrast Slice 1	72

Figure 4.18 CT Head WO Contrast Slice 29	72
Figure 4.19 CT Head WO Scout Slice 1	72
Figure 4.20 CT Head WO Scout Slice 2	73
Figure 4.21 MRA Carotid Left 2D TOF Rotate S1.....	73
Figure 4.22 MRA Carotid Left 2D TOF Rotate S11.....	73
Figure 4.23 XR Chest AP Portable	73
Figure 4.24 Refer image $r(x, y)$, Test image $t(x, y)$, Size $[nx, ny]$	74
Figure 4.25 Illustration of Image Improvement for increased PSNR measurement.....	79
Figure 4.26 Phantom Geometry	83
Figure 4.27 Digital Image Brain 3-D.....	84
Figure 4.28 Digital Image Brain 3-D.....	84
Figure 4.29 Digital Image Brain 3-D.....	84
Figure 4.30 Digital Image Brain 3-D.....	84
Figure 4.31 Digital Image Brain 3-D.....	85
Figure 4.32 Digital Image Brain 3-D.....	85
Figure 4.33 Original Data	86
Figure 4.34 Level 2 Approximation.....	86
Figure 4.35 MATLAB CODE	87
Figure 4.36 MATLAB CODE	88
Figure 6.1 T1 Weighted MR Image	93

List of Images

Image 1.1 Gaussian Noise	6
Image 1.2 Salt and Pepper Noise	7
Image 1.3 Speckle Noise	7
Image 1.4 Brownian Noise	8
Image 2.1 Salt and Pepper Noise – Lena	22
Image 2.2 Denoised with Median Filter – Lena	22
Image 2.3 Original Image - Lena	25
Image 2.4 Noisy Image 10σ - Lena	25
Image 2.5 Noisy Image 20σ - Lena	25
Image 2.6 Noisy Image 40σ - Lena	25
Image 2.7 Denoised Image 10σ - Lena	25
Image 2.8 Denoised Image 20σ - Lena	25
Image 2.9 Denoised Image 40σ - Lena	25
Image 2.10 Original – Darkly Man	28
Image 2.11 Black/White Darkly Man before Convolution	28
Image 2.12 Salt and Pepper Noised Added – Darkly Man	28
Image 2.13 Results of Smoothing Noised Image	28
Image 2.14 Airy Point Spread Function	31

Images

Image 2.15 Lena Image Denoised with $K=0.001$	33
Image 2.16 Lena Image Denoised with $K=0.010$	33
Image 2.17 Lena Image Denoised with $K=0.100$	33
Image 2.18 Lena Image Denoised with $K=1.000$	33
Image 2.19 Lena Image Denoised with $K=5.000$	33
Image 2.20 Original Lena	34
Image 2.21 Noisy Lena	34
Image 2.22 MATLAB Denoised Result	34
Image 2.23 C Program Denoised Result	34
Image 3.1 lena128.pgm	40
Image 3.2 Lena 64 x 64	41
Image 3.3 Lena 32 x 32	41
Image 3.4 Lena 16 x 16	41
Image 3.5 Original Woman Image	47
Image 3.6 Single Stage Woman Image	47
Image 3.7 Second Stage Woman Image	47
Image 3.8 Cameraman	57
Image 3.9 Image Corrupted with Gaussian Noise	57
Image 3.10 Image after Application of VisuShrink	57

Images

Image 3.11 Original Image Woman	57
Image 3.12 Noisy Image Woman with Gaussian Noise	57
Image 3.13 Denoised Image Woman Application of VisuShrink	57
Image 3.14 Cameraman Input Corrupted Gaussian Noise	58
Image 3.15 Cameraman Image after Application SureShrink	58
Image 3.16 Full Moon	60
Image 3.17 Noisy Image Corrupted with Speckle Noise	60
Image 3.18 Decomposed Image	60
Image 3.19 Image Denoised with wavelet db4	60
Image 3.20 Cameraman	61
Image 3.21 Noisy Image Corrupted with Gaussian Noise	61
Image 3.22 Decomposed Image	61
Image 3.23 Image Denoised with wavelet sym4	61
Image 3.24 Woman	62
Image 3.25 Noisy Image Corrupted with Salt and Pepper Noise	62
Image 3.26 Decomposed Image	62
Image 3.27 Image Denoised with wavelet bior 6.8.....	62
Image 4.1 Images of the Crab Pulsar Gamma Spectrum	64
Image 4.2 Images of the Crab Pulsar X-ray Spectrum	64
Image 4.3 Images of the Crab Pulsar Optical Spectrum	64

Images

Image 4.4	Images of the Crab Pulsar Infrared Spectrum	64
Image 4.5	Images of the Crab Pulsar Radio Spectrum	64
Image 4.6	Original Image MRI	76
Image 4.7	Noisy Image MRI	76
Image 4.8	Denoised Image MRI	76

Chapter 1

Chapter 1 Introduction

1.1 Fundamentals in Digital Image Processing

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing [2]. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be model in the form of multidimensional systems.

Digital image processing techniques have become important in a wide variety of fields such as astronomy, computer science, information science, and medicine (MRI scans, Laser Imaging) [18]. The results of research have established the value of image processing techniques in a variety of problems ranging from restoration and enhancement of space/interplanetary probe pictures, the processing of fingerprints, and the forensic use of face recognition algorithms [18].

Many of the techniques of digital image processing were developed in the 1960s at the Jet Propulsion Laboratory, Massachusetts Institute of Technology, Bell Laboratories, University of Maryland, and a few other research facilities, with application to satellite imagery, medical imaging, videophone, character recognition, and photograph enhancement [2]. Image processing has been combined with the Finite Element method to provide a technique of studying the

structural and body mechanics of the human body.

Major areas of image processing are [3]:

- (i) Image Representation
- (ii) Image Transformation
- (iii) Image Enhancement
- (iv) Image Restoration
- (v) Color Image Processing
- (vi) Transform-domain Processing
- (vii) Image Compression
- (viii) Image Representation and Description
- (ix) Object Recognition
- (x) Linear Filtering
- (xi) Principal Component Analysis
- (xii) Hidden Markov models
- (xiii) Anisotropic diffusion
- (ivx) Neural network
- (xv) Partial differentials
- (xvi) Wavelets

Image enhancement techniques are used to highlight certain features of interest in an image. Two important examples of image enhancement are: (i) increasing the contrast, and (ii) changing the brightness level of an image so that the image looks better [22]. It is a subjective area of image processing. On the other hand, image restoration is very much objective. The restoration techniques

are based on mathematical and statistical models of image degradation. Denoising (filtering) and deblurring tasks come under this category.

Image analysis provides a quantitative measurement of certain aspects of an image, such as texture analysis, object detection, crop outlining and recognition, photo-interpretation, or X-ray medical diagnosis [2].

Image processing may be performed in the (1) the **spatial-domain** or in (2) the **frequency transform domain**. Processing techniques in the first category are based on direct manipulation of the pixels in an image [34]. Processing techniques in the second category are based on modifying the Fourier spectrum of an image. Many Frequency domain methods, in addition to the Fourier method are utilized. In certain cases the two methods are combined to achieve a favorable result.

A methodology utilizing the MATLAB Toolbox™ tools demonstrate the comprehensive de-noising algorithms provided for filtering digital images. A comprehensive set of reference-standard algorithms and graphical tools are provided for image processing and analysis, visualization, and algorithm development. You can develop external algorithms for example in C Language, Fortran Language, and Visual Basic Language that can be access as Function Subroutines. Image enhancement, image deblurring, feature detection, noise reduction, image segmentation, geometric transformations, and image registration are all provided within the MATLAB Toolbox™ options.

The technique will take the image of an input image file and perform a filtering process on the graphics file: the input formats that are frequently used are PNG, PGM, TIFF, GIF and JPEG. The PGM format will be used for this study project due to its ease of input. The PGM format is a readable ASCII code, thus one can edit the file with well-known text editors such as Microsoft Word Pad,

Microsoft Word, and the Windows Based VI editor. The differences in Digital Images files are readily seen as numbers between 0 and 255 are demonstrated.

The primary objective of an algorithm is to enhance the image by removing the undesirable effects: **Noise and Blur** are often the type of degradations present in Image Files [33]. The resulting image will therefore be improved as compared to the original image.

1.2 Noise in Digital Images

In this section, various types of noise corrupting an image signal are studied; the sources of noise are discussed, and mathematical models for the different types of noise are presented. Typical images are corrupted with noise modeled with either a Gaussian, uniform, or salt or pepper distribution [43]. Another typical noise is a speckle noise, which is multiplicative in nature. The behavior of each of these noises is described in Section 1.3 through Section 1.4.

Noise is present in an image either in an additive or multiplicative form. An additive noise follows the rule,

$$w(x, y) = s(x, y) + n(x, y), \quad (1.1)$$

while the multiplicative noise satisfies

$$w(x, y) = s(x, y) \times n(x, y), \quad (1.2)$$

where $s(x,y)$ is the original signal, $n(x,y)$ denotes the noise introduced into the signal to produce the corrupted image $w(x,y)$, and (x,y) represents the pixel location. The above image algebra is done at pixel level. By image multiplication, the brightness of the image is varied.

1.3 Gaussian Noise

Gaussian noise is evenly distributed over the signal. This means that each pixel in the noisy image is the sum of the true pixel value and a random Gaussian distributed noise value [37]. As the name indicates, this type of noise has a Gaussian distribution, which has a bell shaped probability distribution function given by,

$$F(g) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(g-m)^2 / 2\sigma^2} \quad (1.3)$$

where g represents the gray level, m is the mean or average of the function, and σ is the standard deviation of the noise [18]. Graphically, it is represented as shown in Figure 1.1. Image 1.1 illustrates the Gaussian noise with mean (variance) as 1.5 over a base image with a constant pixel value of 100 [2].

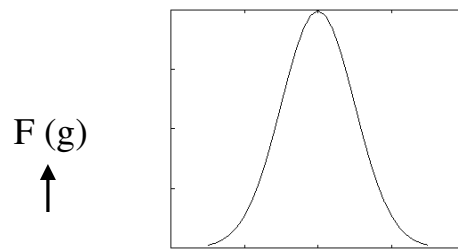


Figure 1.1 Gaussian distribution
g →

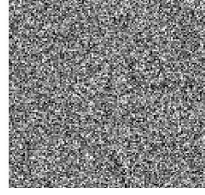


Image 1.1: Gaussian noise (mean=0, variance 0.05)

1.4 Salt and Pepper Noise

Salt-and-pepper noise is a form of noise that presents itself as sparsely occurring white and black pixels. Salt and pepper noise is an impulse type of noise, which is also referred to as intensity spikes. An effective noise reduction method for this type of noise is a median filter [35]. For reducing either salt noise or pepper noise, but not both, a contraharmonic mean filter can be effective [18]. Salt and pepper noise can be caused by errors in data transmission [42]. Also salt and pepper noise can be caused by malfunctioning of pixel elements in the camera sensors or timing errors in the digitization process. It has only two possible values, a and b . The probability of each is typically less than 0.1. The corrupted pixels are set alternatively to the minimum or to the maximum value, giving the image a “salt and pepper” like appearance. Unaffected pixels remain unchanged [42]. For an 8-bit image, the typical value for pepper noise is 0 and for salt noise 255. The probability density function (PDF) for this type of noise is shown in Figure 1.2. Salt and pepper noise with a variance of 0.05 is shown in image 1.2.

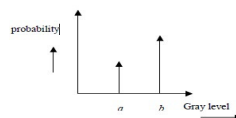


Figure 1.2

PDF for salt and pepper noise

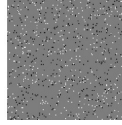


Image 1.2 Salt and Pepper Noise

1.5 Speckle Noise

Speckle noise is a multiplicative noise. This type of noise occurs in almost all coherent imaging systems such as laser, acoustics and SAR (Synthetic Aperture Radar) imagery [18]. The source of this noise is attributed to random interference between the coherent returns. Fully developed speckle noise has the characteristic of multiplicative noise. Speckle noise follows a gamma distribution and is given as

$$F(g) = \frac{g^{\alpha-1}}{(\alpha-1)! a^{\alpha}} e^{-\frac{g}{a}} \quad (1.4)$$

where variance is a^2 α and g is the gray level.

On an image, speckle noise (with variance 0.05) looks as shown in Image 1.3. The gamma distribution is given below in Figure 1.3.

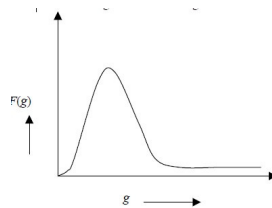


Figure 1.3
Gamma Distribution

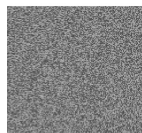


Image 1.3: Speckle Noise

1.6 Brownian Noise Speckle

Brownian noise comes under the category of fractal or $1/f$ noises. The mathematical model for $1/f$ noise is fractional Brownian motion [42]. Fractal Brownian motion is a non-stationary stochastic process that follows a normal distribution. Brownian noise is a special case of $1/f$ noise. It is obtained by integrating white noise. It can be graphically represented as shown in Figure 1.4. On an image, Brownian noise would look like Image 1.4.



Figure 1.4: Brownian noise distribution

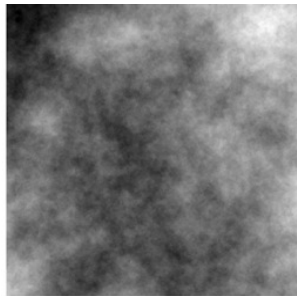


Image 1.4: Brownian noise

1.7 Quantifying Noise

1.7.1 Mathematics

There are three calculations that provide metrics for determining the quality of a denoised image. The terms PSNR, MSE, and MAE are typically calculated in image processing studies. The term PSNR is the peak signal to noise ratio. The term MSE is the mean square error. The term MAE is the mean absolute error. To effectively de-noise an image it is necessary to **increase** the value of PSNR, and **minimize** the values of MSE and MAE.

$$\text{PSNR} = 10\log_{10}\left(\frac{P_s}{P_n}\right) \quad (1.5)$$

P_s =Power of Signal

P_n =Power of Noise

the MSE (Mean Squared Error) is express as:

$$\text{MSE} = \frac{1}{n^2} \sum_{x,y=1}^n (z(x,y) - s(x,y))^2, \quad (1.6)$$

$z(x,y)$ is the estimate of the signal and $s(x,y)$ is the original signal without noise and n is the size of the signal

MAE =Original Signal - De-noised Image

An 8-bit image has values ranging between 0 and 255.

For PSNR calculations, the numerator is 255^2 in all cases.

The PSNR is usually expressed in terms of the logarithmic decibel scale.

The higher the PSNR, the better degraded image has been reconstructed to match the original image and the better the reconstructive algorithm [20]. In image processing, the **PSNR must be greater than about 20 dB** to be considered a **high-quality picture** [2]

1.8 Objectives

The basic focus of this thesis is the estimation of the uncorrupted image from the distorted or noisy image, and is referred to as image “denoising”. There are several methods that can improve the restoration of an image from noisy distortions. Selecting the appropriate method plays a major role in getting the desired image. Denoising methods are often problem specific. For example, a method that is used to denoise satellite images, astronomy images, or micro blood vessels images may not be suitable for denoising medical images such as MRI scans and CT scans. In this thesis, a study is made on the various denoising algorithms and each is implemented in either Matlab6.5 [31] or C code subroutines. The methods are compared and classified in terms of definable metrics (PSNR, MSE, MAE). In order to quantify the performance of the various denoising algorithms, a high quality image is taken and some known noise is added to it. This would then be given as input to the denoising algorithm, which produces an image close to the original high quality image. Two popular

techniques are studied in this thesis: Wiener and Adaptive Wiener filter, Wavelet Techniques. The main advantage of wavelet basis is that they despite having irregular shape are able to perfectly reconstruct functions with linear and higher order polynomial shapes, such as, rectangular, triangle, 2nd order polynomials, etc. Note that Fourier basis fail to do so, as in case of the famous example of the rectangular function at the edges. As a result, wavelets are able to denoise particular images far better than conventional filters that are based on Fourier transform design and that do not follow the algebraic rules obeyed by the wavelets.

1.9 Conclusion

Chapter I introduces some basic concepts of digital image processing. The fundamentals of digital image processing, sources of noise and types of noise in an image are discussed. The existing filters and their merits and demerits and the quantifying image metrics are discussed. In Chapter II, image enhancement techniques are reviewed and applied to example problems. Then in Chapter III, wavelet theory and analysis are applied to the processing of several images and the results are discussed. The implementation of MATLAB for Wavelet analysis is presented. Detailed examples are presented in the selection of Wavelet Thresholding in Chapter III. Chapter IV examines the application of image processing in the fields of Science and Medicine. Several Wavelet types are studied for effectiveness. The importance of Tomography in the medical field is examined. Examples of MRI and X-ray scans are shown. Chapter V summarized the conclusions. The Scope of Future applications of Digital Image Processing are discussed in Chapter VI.

Chapter 2

2.1 Digital Image

A digital image is a collection of picture elements, or pixels, usually arranged in a rectangular array. A pixel is the smallest graphic unit of the image and, as an atom, it is not further divisible [18]. It is assigned one or more numerical values, which define its appearance (color, brightness, etc.). The position of each pixel in the image is given by its co-ordinates, which specify horizontal and vertical location. They can also be interpreted as column and row and follow the axis sign and conventions illustrated in Figure 2.1 – the pixel of co-ordinates (0, 0) being in the upper left corner of the image [18].

Different classes of images exist, based on the numerical values associated with pixels: binary, grayscale, color, etc. In binary (or *bitmap*) images each pixel contains just one bit of information, which can therefore represent two statuses, 0 and 1. The image can be viewed in black and white, where black corresponds to the value 1 and white to the value 0. Alternative representations, however, are possible: the same image can be viewed, for instance, in red and white, in blue and red or in white and black (Figure 2.2) [18]. The amount of information contained in each case is the same, although the way in which this information is displayed changes. This depends on the way in which numerical values are put into correspondence with colors – or, in more precise terms, on the selected colormap: a colormap associates color with the numerical values of pixels, allowing an image to be shown on a computer screen or printed on a page.

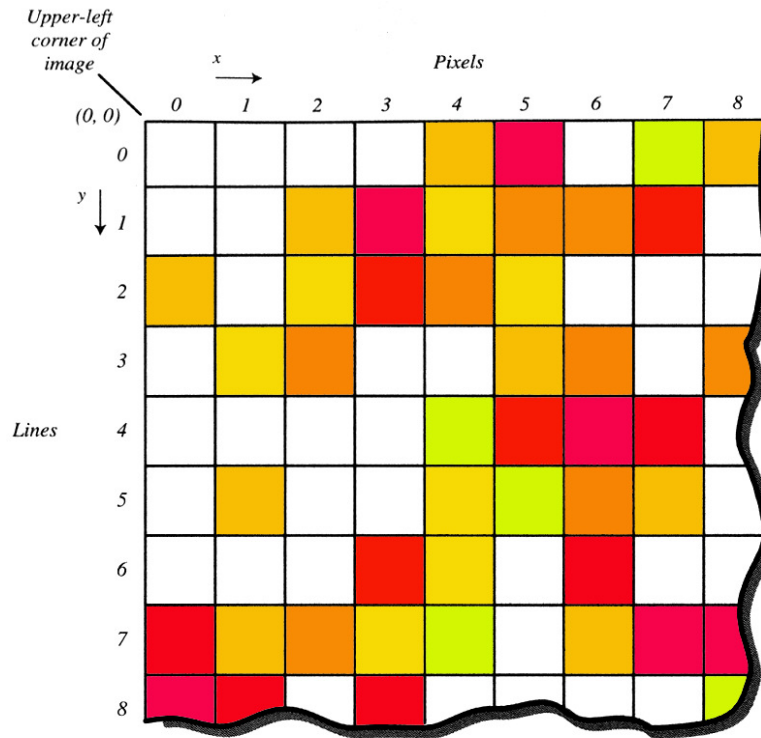


Figure 2.1
The discrete pixel numbering convention
Image Adapted from [3]



Figure 2.2
Using different colormaps to display the same binary image: black and white, red and white, blue and red, white and black (which is the inverse of black and white [2]).

Most of the figures presented in this work belong to the class of images called *greyscale*. In greyscale images each pixel can take more than two statuses: generally 8 bits of information are used, which correspond to 256 ($=2^8$) possible values [18]. Each of them is put into correspondence with a grey level using a colormap (Figure 2.3).

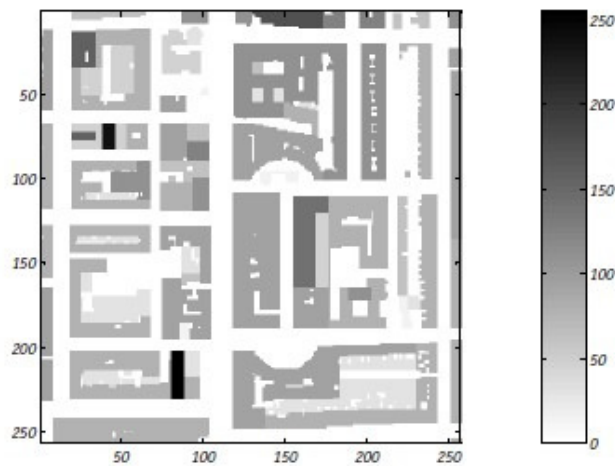


Figure 2.3

A greyscale image [2]

Each grey level represents a value between 0 and 255.

As in the case of binary images, colormaps can be arbitrarily defined. The default representation of a greyscale image in Matlab and in most image-processing software assigns black to 255, white to 0 and different tonalities of grey to the values in between, as shown in Figure 2.3 [19]. Other options are also available, opening up a wide range of representational possibilities: a number of them, which are defaults in Matlab [31], are shown in Figure 2.4.

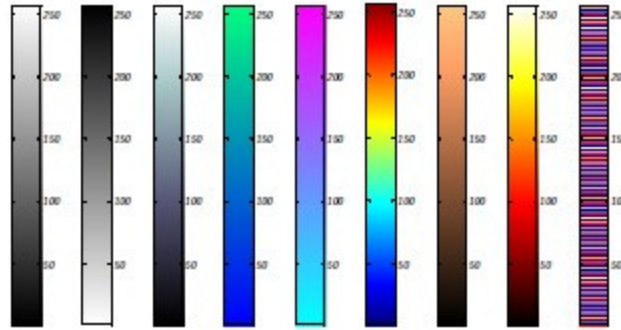


Figure 2.4

Default Color Maps available in Matlab [31]

Image processing software allows the creation of user-defined colormaps, whose ad-hoc properties can be used to emphasize certain features of an image (for instance, highlighting pixels in a certain numerical range) [19]. Grayscale images can be used to display color scenes with appropriate colormaps. Their ability to match real world colors is limited to the amount of information stored in each pixel (one byte, which, as it has been said, represents values between 0 and 255). Therefore, full color images are usually dealt with by tripling the number of channels: three images of 8 bits per pixel each are used, defining the amount of Red, Green and Blue contained in each pixel. These images are called *RGB* and can accurately represent most real colors (they allow the generation of over 16 million colors, which correspond to $256 \times 256 \times 256$ combinations of red, green and blue).

RGB images use 3 times more computer memory than greyscale ones, which in turn, which in turn use 8 times more than bitmaps (8 bits versus 1) [18]. An RGB image contains more information than a greyscale, which in turn contains more information than a bitmap. Going from RGB to the other formats

results in discarding part of the digital information. Bitmap, greyscale, and colormap are important parameters of an image. The resolution defines the way in which the image is acquired. A digital image for example could be either 100x100 or 1000x1000 pixels. The number of pixels per inch defines the image resolution and is called *dpi* (dots per inch). It is a fundamental parameter which characterizes the digitization process – i.e. the process by which an analogue signal in a 2-D continuous space is converted into a 2-D discrete space through a sampling process. The higher the resolution and the number of pixels used, the better the quality of the image.

2.2 Processing Methods

Spatial domain methods and Frequency domain methods are utilized to filter noisy images. Several of the existing techniques developed for image denoising are listed in Table 2.1 [43].

Spatial Domain Methods	Transform Methods
Mean Filter	Earlier Smoothing Filters (Frequency Domain)
Standard Median	Gaussian Filter
Adaptive Median	Discrete Cosine Transform (DCT)
Bilateral Filter	Fast Fourier Transform (FFT)
Non-Local Means (NLM)*	Wiener*
	Wavelet – Multi Resolution*
	Block Matching 3D Transform(BM3D)*
	Adaptive Principal Component Analysis*
	Steerable Filters*

* High Interest

Table 2.1
List of Important Denoising Techniques in the Spatial
Domain and in the Frequency Domain

The Table lists a summary of the Spatial Domain and Frequency Domain methods. The Table also identifies the methods with high/current research interest.

An integrative procedure is utilized for improving the noise removal of the digital image. Successive application of multiple techniques will improve the total process. Several different approaches exist in modifying a digital image.

2.3 Spatial Domain Methods

2.3.1 Mean Filter

A mean filter acts on an image by smoothing it; that is, it reduces the intensity variation between adjacent pixels [5]. The mean filter is a simple sliding window spatial filter that replaces the center value in the window with the average of all the neighboring pixel values including it [2]. By doing this, it replaces pixels that are unrepresentative of their surroundings. It is implemented with a convolution mask, which provides a result that is a weighted sum of the values of a pixel and its neighbors. It is also called a linear filter. The mask or kernel is a square. Often a 3×3 square kernel is used. If the coefficients of the mask sum up to one, then the average brightness of the image is not changed. If the coefficients sum to zero, the average brightness is lost, and it returns a dark image. The mean or average filter works on the shift-multiply-sum principle [36]. This principle in the two-dimensional image can be represented as shown below (refer to Figure 2.5).

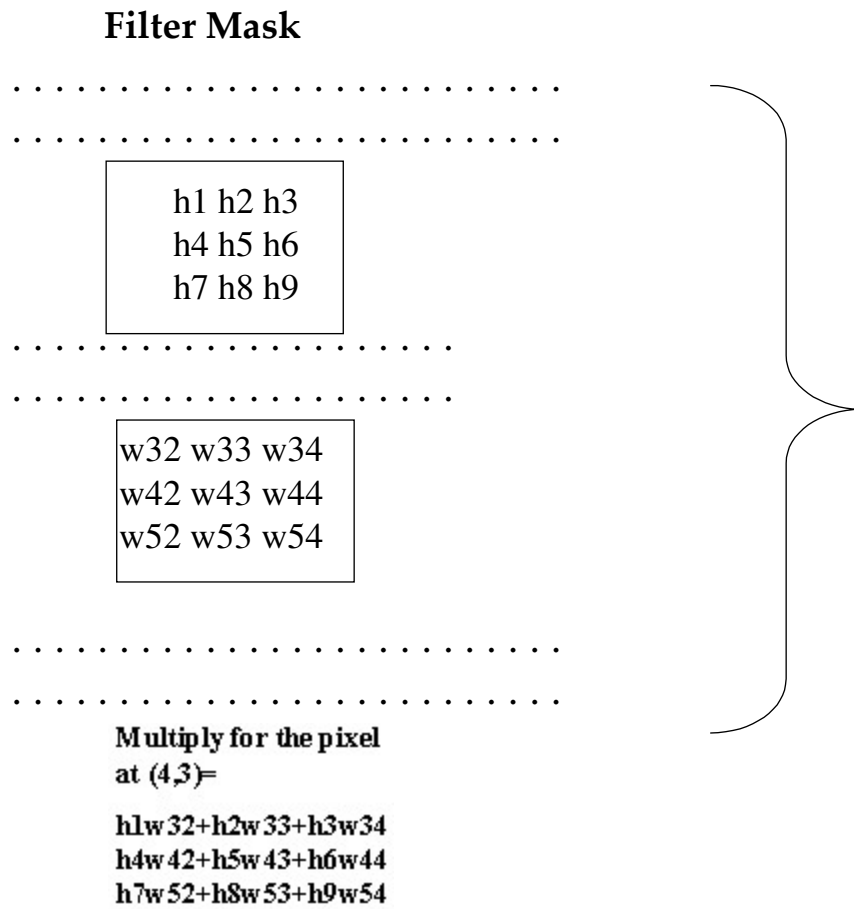


Figure 2.5
The Shift-Multiply-Sum Principle [7]

2.3.2 Median Filter

The standard median filter is a nonlinear filtering technique used to remove noise in the spatial domain [2]. Noise reduction can be a pre-processing step to improve the results of later processing (for example, edge detection on an image [18]).

The main idea of the median filter is to run through the image entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value (Figure 2.6).

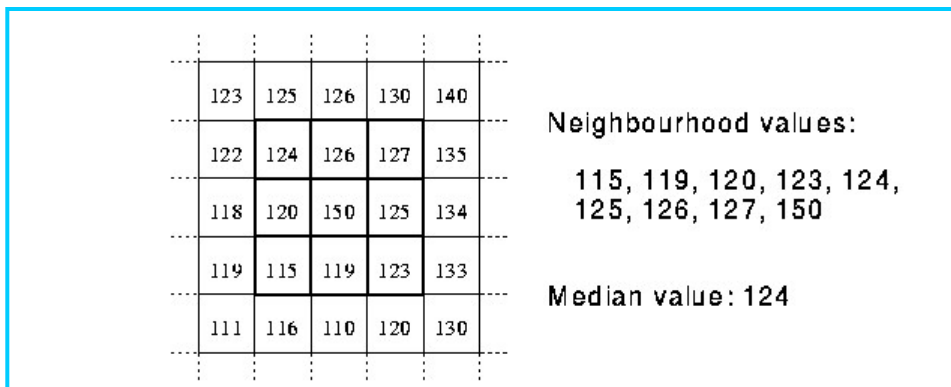


Figure 2.6

Illustrates an example calculation [7]

Median filtering is very good when the noise produces extreme outlier pixel values, as for instance in salt and pepper noise [37]. The median filter has been incorporated into the

C code written for this report. Using this code, an example of the filtering of salt and pepper noise is show below.



Image 2.1

Salt and Pepper Noise



Image 2.2

Denoised with Median Filter

The image in Image 2.2 is improved with the removal of the salt and pepper noise. The metrics are shown below in Table 2.2.

Image Processing Metrics For Median Filter	
PSNR	18.1583
MSE	0.1202
MAE	0.104

Table 2.2
Metrics for Denoised Lena Image
with Median Filter

2.3.3 Non-local Means

Local smoothing methods and the frequency domain filters aim at a noise reduction and at a reconstruction of the main geometrical configurations but not at the preservation of the fine structure, details, and texture [5].

Due to the regularity assumptions on the original image of earlier smoothing methods, details and fine structures are smoothed out because they behave in functional aspects as noise [5].

The NL-means algorithm takes advantage of the high degree of redundancy of any natural image [5]. Unlike "local mean" filters, which take the mean value of a group of pixels surrounding a target pixel to smooth the image, non-local means filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. These results in much greater post-filtering clarity and less loss of detail in the image compared with local mean algorithms.

Every small window in a natural image has many similar windows in the same image. For windows close by, at one pixel distance, local regularity assumption is valid. One can define as "neighborhood of a pixel i " any set of pixels j in the image such that a window around j looks like a window around i . All pixels in that neighborhood can be used for predicting the value at i .

The NL-means algorithm estimates the value of x as an average of the values of all the pixels, whose Gaussian neighborhood looks like the neighborhood of x ,

$$NL(v)(x) = \frac{1}{C(x)} \int_{\Omega} e^{-\frac{(G_a * |v(x+.) - v(y+.)|^2)(0)}{h^2}} v(y) dy, \quad (2.1)$$

where G_a is a Gaussian kernel with standard deviation a , h acts as a filtering parameter, and

$$C(x) = \int_{\Omega} e^{-\frac{(G_a * |v(x+.) - v(y+.)|^2)(0)}{h^2}} dz \quad (2.2)$$

is the normalizing factor [5].

A C program implementation of the non-local means denoising routine is utilized to denoise the Lena image with varying levels of σ , where σ denotes the standard deviation of the noise. The images of the noisy and denoised Lena case study are shown below. The PSNR performance decreases for increase sigma Noise input. PSNR values of 20 db and above are desired for successful image denoising. The 40 σ noise case shows a result less than desirable.

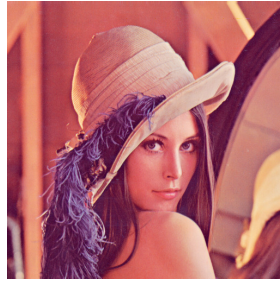
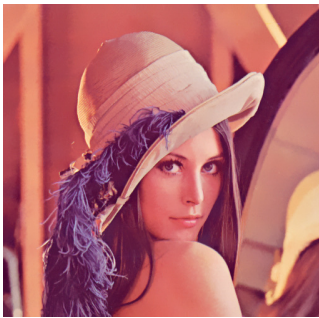
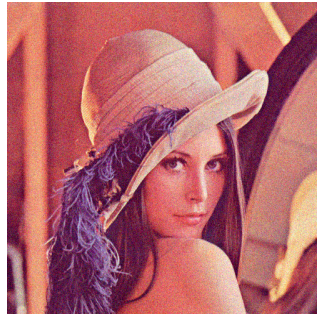
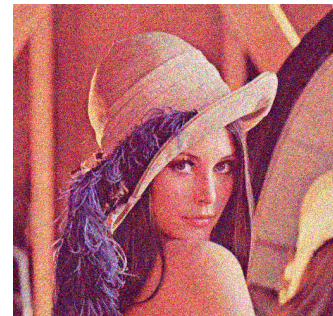


Image 2.3 Original Lena Image

Image 2.4 Noisy 10 σ Image 2.5 Noisy 20 σ Image 2.6 Noisy 40 σ Image 2.7 Denoise 10 σ

PSNR = 29.5171

MSE = 0.0343

MAE = 0.0138

Image 2.8 Denoised 20 σ

PSNR = 23.3267

MSE = 0.0696

MAE = 0.0189

Image 2.9 Denoised 40 σ

PSNR = 17.4303

MSE = 0.1399

MAE = 0.0260

2.4 Frequency Domain Filters

In the Frequency Domain, Selective Frequency Domain Filters are implemented to reduce the noise and enhanced the frequencies wanted. Low pass filters and high pass filters can be used to show some details in the image while hiding other details [3]. Low pass filters blur the image which leads to noise reduction while high pass filters sharpens some image details, such as edges.

2.4.1 Butterworth Filter

A version of the smoothing/sharpening filter is the Butterworth filter. An advantage of the Butterworth filter is that one can control the sharpness of the filter with the order [42]. The equation for the Butterworth high pass filter:

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_o]^{2n}} \quad (2.3)$$

A Butterworth high pass filter keeps frequencies outside the radius D_o and discards values inside. It has a gradual transition from 0 to 1 to reduce ringing artifacts. The Butterworth filter has an **order n** and a **cutoff frequency D_o** .

2.4.2 Convolution - Gaussian Filter

The Gaussian filter is known as a 'smoothing' operator, as its convolution with an image averages the pixels in the image, affectively decreasing the difference in value between neighboring pixels [15]. In the case of Gaussian filtering, the frequency coefficients are not cut abruptly, but a smoother cut process is used instead. This also takes advantage of the fact that the Discrete Fourier Transform of a Gaussian function is an also a Gaussian function. The Gaussian high pass filters attenuates frequency components that are near the image center. The Gaussian high pass filter is given by:

$$H(u, v) = -e^{-D^{2(u,v)/2 \cdot D_0^2}} \quad (2.4)$$

where the filter radius D_0 is in statistics the standard deviation sigma.

In image processing, many of the filter operations are applied to an image by performing a special operation called convolution with a matrix called a kernel [15]. Convolution can be described as a function that is the integral or summation of two component functions, and measures the amount of overlap as one function is shifted over the other.

Gaussian and Sobol filters are applied with a MATLAB routine to demonstrate the effect of reducing noise in images, and improving the quality of images [26]. The MATLAB function **CONV** performs a convolution operation on the input image file Darkly.jpg [4]. An example of the convolution operation is shown in Images 2.10 thru 2.13.

See Appendix A: MATLAB Denoise.

Convolve the Input Image with the Sobel Filter.

Results from the Sobel [26] filter are shown below.

Matlab Files:

sobel2.m (Appendix A)

sobel.m (Appendix A)



Image 2.10 Original
Darkly.jpg



Image 2.11 Black/White
before Convolution

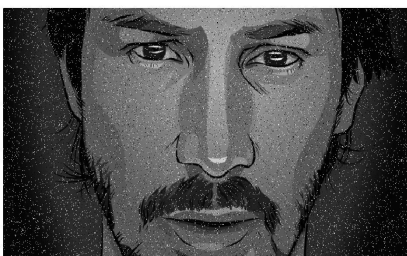


Image 2.12
Salt and Pepper Noise
added

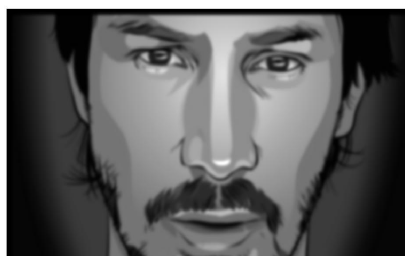


Image 2.13 Results of
Smoothing Noised
Image

2.4.3 Wiener Filter

Norbert Wiener proposed the concept of Wiener filtering in the year 1942 [35]. There are two methods:

- i) Fourier-transform method (frequency-domain)
- ii) Mean-squared method (spatial-domain)

for implementing the Wiener filter. The former method is used for complete restoration (denoising and deblurring) whereas the later is used for denoising. In the Fourier transform method of Wiener filtering, a priori knowledge of the power spectra of noise and the original image is required. Wiener filter is based on the least-squared principle, i.e. the filter minimizes the mean-squared error (MSE) between the actual output and the desired output [40]. The Wiener filter when used for deblurring an image requires the input of the Point Spread Function (PSF) [46].

The Convolution theorem for the continuous case is applied.

If $h(t)$ and $g(t)$ are two functions and $H(f)$ and $G(f)$ are their corresponding Fourier Transforms, then the convolution is defined as

$$g * h = \int_{-\infty}^{\infty} g(\tau)h(t - \tau)d\tau \quad (2.5)$$

where $g(t) * h(t)$ is in the time domain [25]. The convolution theorem is given by:
 $F \{ (g * h)(t) \} = G(\omega)H(\omega)$. The Fourier transform of the convolution is the product of the individual Fourier transforms.

The Wiener filter is used for deblurring an image in the case when the blur kernel (point spread function) is known [46]. The Fourier transform of the deblurred image F is:

$$F = (G/H) * (|H|^2 / (|H|^2 + K)) \quad (2.6)$$

where

G = Fourier transform of original blurry image

H = Fourier transform of blur kernel

k = deblurring parameter ($k > 0$)

The Wiener filter can takes the equivalent form below:

$$F = \left[\frac{H^*}{|H|^2 + \frac{S_n}{S_f}} \right] * G \quad (2.7)$$

G = Fourier transform of original blurry image

H = Fourier transform of blur kernel

H* = Complex conjugate of H

$|H|^2 = H^* * H$

$S_n = |N|^2$ = Power Spectrum of the noise

$S_f = |F|^2$ = Power Spectrum of undegraded image

Setting the value of k is the hard part. The value should depend on the amount of noise expected in the image. If an MxN image has Gaussian white noise with variance σ^2 , then set $k = MN \sigma^2$.

In general, the variance of the noise is not known and one will have to estimate it. Note that if the image has no noise (blur only), then k is set to zero ($k=0$).

Equation (2.7) is referred to as the Wiener Filter or as the minimum mean square error filter [18]. The restored image in the spatial domain is given by the inverse Fourier Transform of the frequency-domain estimate F .

2.4.3.1 Point Spread Function - Wiener Filter

The application of the Wiener Filter is demonstrated with the C code written below. The point Spread Function (PSF) was created in MATLAB and then implemented in the C code to perform the Wiener filtering. The PSF is created with a C program and the AIRY.pgm file is saved for future use.

A plot of The Point Spread Function is shown below.

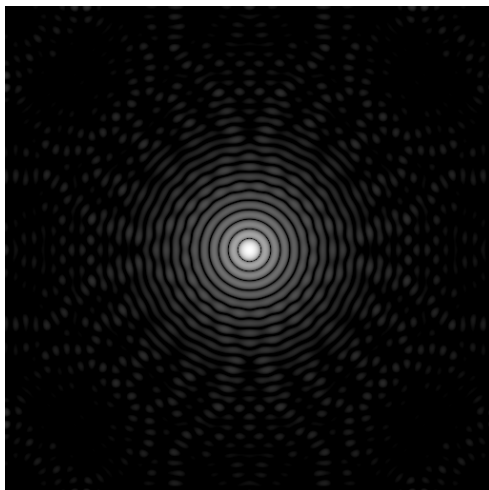


Image 2.14 Airy Point Spread Function

See Matlab code for implementing this function (Appendix A).
airy.m

See Appendix A: MATLAB Denoise.

MATLAB CODE - AIRY POINT SPREAD FUNCTION

2.4.3.2 Application of Wiener Filter

The C code for implementing the Wiener Filter is listed below.

The input consists of the noisy image (lena_noise.pgm) and the Point Spread Function PSF.pgm.

Wiener:

```
im1 = Input_PBM ("lena_noise.pgm");
p   = Input_PBM ("PSF.pgm");

nr  = im1->info->nr; nc = im1->info->nc;

image_fftoc (im1, &fft1);
normalize_set ();
image_fftoc (p,  &fft2);
normalize_clear ();
```

[**Note:** The Power Spectrum ratio $\frac{S_n}{S_f}$ is indicated by the constant k]

$$F = \left[\frac{H^*}{|H|^2 + K} \right] * G \quad (2.8)$$

```
printf ("Enter K: ");
```

The value of the constant k is determined experimentally/interactively to give the best visual result. The Wiener C code is executed interactive to determine the optimum K value. Results for k Values of 0.001, 0.01, 0.10, 1.0, and 5.0 are shown below.



Image 2.15

 $K=0.001$

Image 2.16

 $K=0.010$

Image 2.17

 $K=0.100$ 

Image 2.18

 $K=1.00$ 

Image 2.19

 $K=5.00$

The optimum processing is shown in Image 2.17 (with $K=0.100$). Iterations of the selected values will refine the results.

The MATLAB software provides a built-in function for performing a Wiener filter on a noisy image [31].

The command takes the following form:

$J = \text{WIENER2}(I, [M \ N], \text{NOISE})$ filters the image I using pixel-wise adaptive Wiener filtering, using neighborhoods of size M -by- N to estimate the local image mean and standard deviation

A comparison is made of the C program Wiener result and the image obtained from the built-in MATLAB routine.



Image 2.20
Original Lena



Image 2.21
Noisy Lena



Image 2.22
MATLAB Result
Denoised



Image 2.23
C Program
Denoised

2.5 Algorithm

Filters are written in the C programming Language to perform the operations listed from item 0 through 12. The image files are created in ASCII PGM format.

Spatial Domain and Frequency Domain Filters are coded into the method. Included in this project are the following Filters:

0. Spatial Median Filter
1. Ideal Low Pass
2. Ideal High Pass
3. Band Boost Filter
4. Butterworth Low Pass
5. Butterworth High Pass
6. Exponential Low Pass
7. Exponential High Pass
8. Gaussian Filter
9. Wiener Filter
10. Adaptive Wiener Filter
11. Convolution
12. Unit 1 Pass

The C code listing is listed in the Appendix C: Programs in C Language.

Chapter 3 Wavelets**Chapter 3****3.1 Introduction**

Over the past decade, wavelet transforms have received a lot of attention from researchers in many different areas. There are two main types of wavelet transform: continuous and discrete. Both discrete and continuous wavelet transforms have shown great promise in diverse fields such as image compression, image de-noising, signal processing, computer graphics, and pattern recognition [37].

Computer programs will often use the discrete wavelet transform. The discrete transform is very efficient from the computational point of view. In this paper, the calculation and modeling of the wavelet transform coefficients are studied.

Denoising involves reconstituting the signal as well as possible on the basis of the observations of a useful signal corrupted by noise. The methods based on wavelet representations yield simple algorithms that are often powerful and easier to work with than traditional methods of function estimation [39].

3.2 Wavelet Transforms

3.2.1 Preview of Wavelet Decomposition

The wavelet representation of an image usually consists of a few large coefficients and lots of small ones. The image is well approximated by a small set of its largest coefficients [29]. The wavelet representation of Gaussian noise is more evenly distributed. This is because an orthonormal transformation of Gaussian noise gives back Gaussian noise. For a noisy image, setting to zero the large pool of small wavelet coefficients removes much of the noise while the few retained large coefficients still give a good approximation of the noiseless image [33]. This is a major attraction of the Wavelet method. The large coefficients still contain a bit of noise but most of their amplitude is a useful signal. A typical signal or image of interest is difficult to define, or model, positively. But it generally possesses the following non-properties [44]:

- 1) -it is not a pure noise, i.e. it has some "regular" structures;
- 2) -it is not fully regular/stationary, e.g. totally flat or like a "constant" sine wave.

Those could be considered as "constant", and not conveying a lot of information (or a non-changing information) [29].

Instead, it may possess a certain quantity of time- or space-limited, varying structures. It is somewhat unstationary, but not wildly, unless it might appear too cluttered to be distinguished from noise.

Among the time-space-varying structures, one sometimes distinguishes four types of features:

- 1) -very smooth ones, long term trends, shadows, close to low-degree polynomials;
- 2) -wiggling ones, over a medium distance, like short oscillations, periodic textures
- 3) -sharp ones, like a jump, or edges.
- 4) -unmodeled patterns and noise.

Wavelets are the building blocks that are able to quickly de-correlate typical signals or images [39]. Features get better concentrated when convolved with wavelets. If the wavelets preserve energy (are orthogonal or close to), they will turn a gaussian noise into something close to gaussian. The noise will not be better concentrated, and sticks to "low coefficients". With wavelets, interesting features (to the eye) in images get better concentrated and easier to distinguish from the noise floor than in the original data. The science of wavelet denoising consists in separating the nice features from the bad noise by:

- 1) -choosing the appropriate wavelet transform to best concentrate the features in your data,
- 2) -choosing the appropriate threshold to cut right between the high and the low coefficients.

Thus the wavelet projection of noise elements has much lower magnitude as compared to true data, or signal of interest. When a threshold is applied on these projections, mostly the noise coefficients get zeroed out. With the inverse transform, the footprint of the noise coefficients do not show in time domain reconstruction, thereby de-noising the signal.

The wavelet method is an example of one type of image transform. Just as the Fourier transform is a special case of an image being decomposed into the sum of weighted sine and cosine components [37]. There are in fact many functions that may be used for decomposition. The Fourier transform and Wavelet transform being examples of a large class of transforms that have the following form [37]:

$$Trans(u, v) = \sum_{x=0}^N \sum_{y=0}^N f(x, y)g(x, y, u, v) \quad (3.1)$$

where f is the two-dimensional function being transformed, and g is called the *kernel* of the transformation, wherein reside the basis functions.

For the Fourier transform, the kernel is:

$$g(x, y, u, v) = \frac{e^{-2\pi j(ux+vy)/N}}{N} \quad (3.2)$$

this is the sum of the sine and the cosine terms.

The idea of an image transform is a general one with the Fourier and Wavelet transforms being special cases. Wavelets provide advantages over Fourier transforms. They do a good job in approximating signals with sharp spikes or signals having discontinuities. Wavelets can also model speech, music, video and non-stationary stochastic signals [42]. Wavelets can be used in applications

such as image compression, turbulence, human vision, radar, earthquake prediction, etc. [2].

3.2.2 Simplest Example of Image Transform-Walsh Transform

The simplest example of an image transform is the **Walsh** transform, which uses the values 1 and -1 as the basis of the series expansion [37].

The transformation kernel for the two-dimensional case is:

$$g(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{nbits-1} -1^{\text{bit}(i,x)\text{bit}(n-i-1,u)+\text{bit}(i,y)\text{bit}(n-i-1,v)} \quad (3.3)$$

where the function $\text{bit}(i, x)$ returns the value of bit number i of the number x .

This transform can be used as a primitive compression tool. A C program (see appendix A) is utilized to demonstrate the application of the Walsh transform.

The image lena128.pgm is reconstructed using all of the pixels.

A program walsh.c finds the Walsh transform of the 128 x 128 image.



Image 3.1 lena128.pgm

128 x 128

The same image is shown but now reconstructed from a subset of the 128 x 128 pixels: 64 x 64, 32 x 32, and 16 x 16.



Image 3.2

64 x 64



Image 3.3

32 x 32

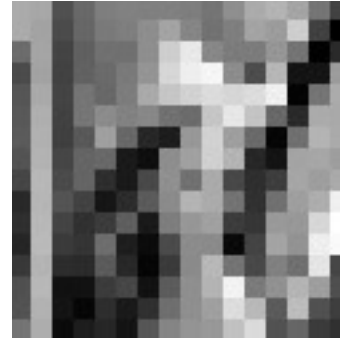


Image 3.4

16 x 16

The quality goes down as the number of pixels used in the reconstruction goes down. The Walsh transform show that a simple function, even a simple numeric constant can be used as a basis for a transformation of this type [17]. A wavelet transform differs from a Fourier transform in that both a **frequency factor** and a **scaling factor** are associated with it. The wavelets are generated from a single basic wavelet $\psi(t)$, the so-called mother wavelet, by scaling and translation. The scaled and translated wavelet is written as:

$$w_{s,t}(x) = g\left(\frac{(x - t)}{s}\right) \quad (3.4)$$

where s and t refer to a scaling and translation of the wavelet g .

For a given function of sampled data $f(x)$, the wavelet decomposition can be written:

$$f(x) = \sum_{i=0}^S \sum_{j=0}^T c_{i,j} w_{i,j}(x) \quad (3.5)$$

which is a linear combination of the $w_{i,j}$ functions. The $C_{i,j}$ are wavelet coefficients, and their values represent the relative contribution of that specific wavelet in the decomposition. The wavelet decomposition of f results in a two-dimensional function of s and t .

3.2.3 The Haar Wavelet

The Haar wavelet is also a simpler type of wavelet, and is also one of the oldest wavelet types. It has been used in image analysis for many years as the Haar transform [37]. It also strongly resembles the Walsh basis, because the Haar wavelet is a step function taking the values +1 and -1. The simplest Haar wavelet is defined as:

$$\psi_H(t) = \begin{cases} +1 & 0 < t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$

A linear combination of functions can be constructed whose sum approximates the required function:

$$f(t) = \sum_{j=-\infty}^{\infty} A_j \sum_{k=-\infty}^{\infty} B_{jk} \psi_H(2^j t - k) \quad (3.7)$$

A sample collection of these wavelets as graphs is shown in Figure 3.1 [37].

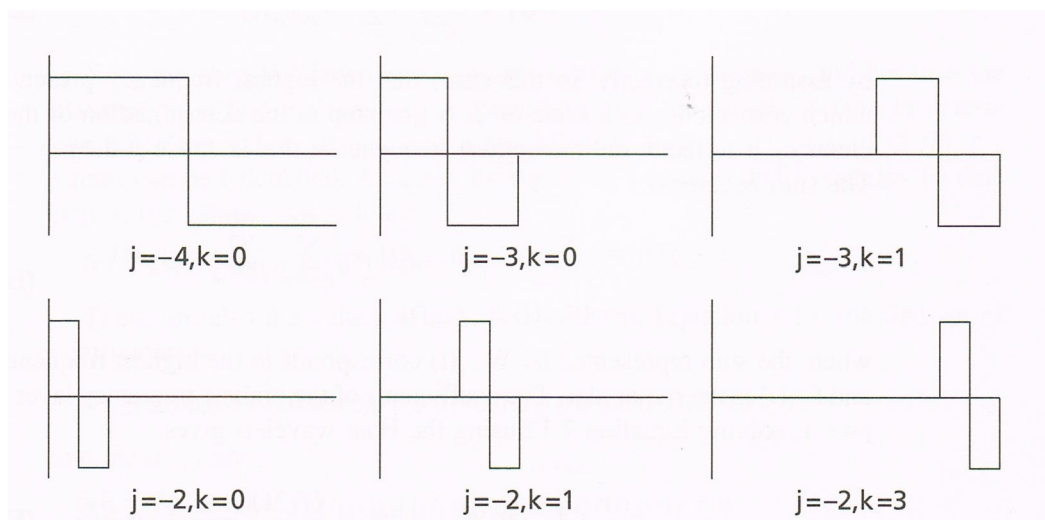


Figure 3.1

Haar wavelets of various scales and translations

They have the appearance of square waves, scaled and translated over a small range.

3.3 Image Denoising

Denoising is a procedure to recover an image that has been corrupted by noise. After discrete wavelet decomposition the resulting coefficients can be modified to eliminate undesirable image components. To implement wavelet thresholding a wavelet shrinkage method for de-noising the image is executed [6]. Denoising consists of decomposing the observed image into wavelets and using thresholds to select which coefficients are retained, from which an improved image is synthesized. The image denoising algorithm is summarized in the following steps [8]:

Algorithm: Wavelet Image Denoising

- **Choice of a wavelet (e.g., Haar, Symmlet, etc) and number of levels or scales for the decomposition. Computation of the forward wavelet transform of the noisy image.**
- **Estimation of a threshold.**
- **Choice of a shrinkage rule and application of the threshold by hard or soft thresholding.**
- **Application of the Inverse transform (wavelet reconstruction) using the modified (threshold) coefficients.**

The Algorithm (Wavelet Image De-noising) is expressed in Figure 3.2.

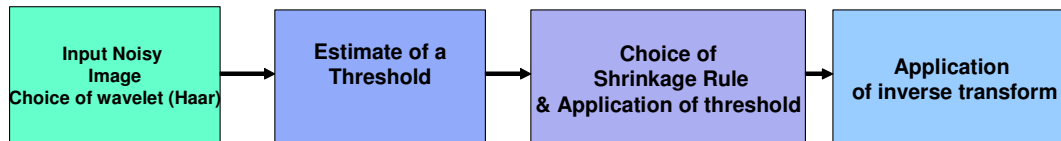


Figure 3.2: Block Diagram of Image denoising using Wavelet transform

3.4 Discrete Wavelet Transform

The discrete wavelet transform (DWT) is an implementation of the wavelet transform using a discrete set of wavelet scales and translations obeying definite rules. The transform decomposes the signals into mutually orthogonal set of wavelets.

The orthonormal basis or wavelet basis is defined as [39]:

$$\psi_{(j,k)}(x) = 2^{-j/2} \psi(2^j x - k) \quad (3.8)$$

The scaling function is given as [39]:

$$j_{(j,k)}(x) = 2^{-j/2} j(2^j x - k) \quad (3.9)$$

where ψ is called the wavelet function and j and k are integers that scale and dilate the wavelet function. The factor ' j ' in Equations (3.8) and (3.9) is known as the scale index, which indicates the wavelet's width. The location index k provides the position. The wavelet function is dilated by powers of two and is translated by the integer k . This transform decomposes the signals into mutually orthogonal set of wavelets. The implementations of the DWT algorithm include Haar, Daubechies, Symlets and Coiflets orthogonal wavelets [23]. There are several ways of implementing the DWT algorithm. The most known one is the Mallat-algorithm or Mallat-tree decomposition [38]. In this algorithm, the DWT is computed by successive low-pass and high-pass filtering of the time-domain signal. The original signal is then obtained by concatenating all the coefficients starting from the last level of decomposition. The wavelet decomposition of an image is done as follows: in the first level of decomposition, the image is decomposed into four subbands, namely HH (high-high), HL (high-low), LH (low-high) and LL

(low-low) subbands. The HH subband gives the diagonal details of the image; the HL subband gives the horizontal features while the LH subband represents the vertical structures. The LL subband is low resolution residual consisting of low frequency components and it is further split at higher levels of decomposition. It is convenient to label the subbands of transform as shown in Figures 3.3, 3.4, and 3.5 [38].

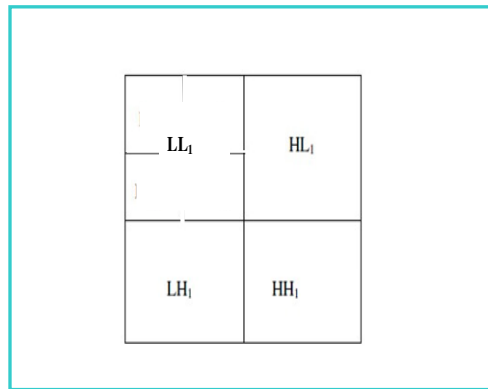


Figure 3.3

First Level Decomposition

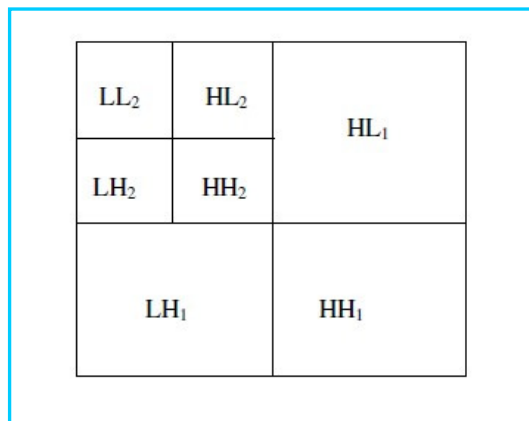


Figure 3.4

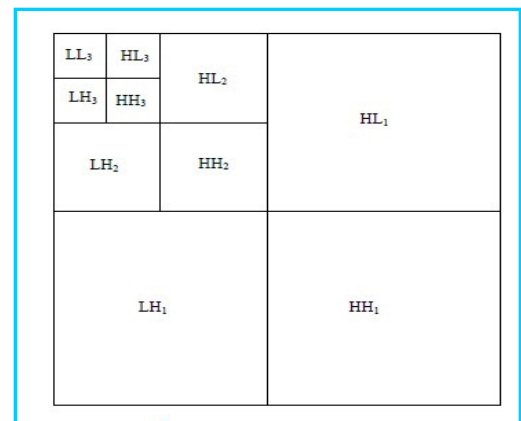
Second Level Subbands
of the 2D Wavelet Transform DWT

Figure 3.5

Third Level Subbands
of the 2D Wavelet Transform DWT

The Woman image is used to show an example of the decomposition of an image in subbands. The original image is displayed followed by Single Stage decomposition and Two Stage decomposition.



Image 3.5 Original Woman Image

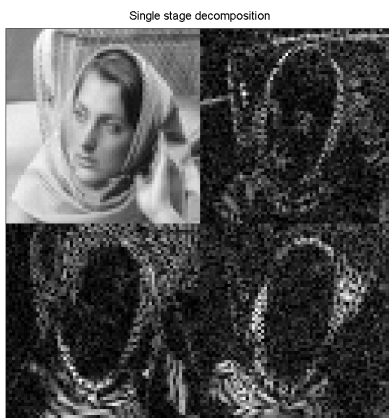


Image 3.6 Single Stage Decomposition



Image 3.7 Second Stage Decomposition

3.5 Wavelet Thresholding

Donoho and Johnstone [16] did much of the work on filtering of additive Gaussian noise using wavelet thresholding. From their properties and behavior, wavelets play a major role in image compression and image denoising. Image denoising is a major application of wavelet analysis. Wavelet coefficients calculated by a wavelet transform represent change in the time series at a particular resolution. By considering the time series at various resolutions, it is then possible to filter out noise.

The term wavelet thresholding is explained as decomposition of the data or the image into wavelet coefficients, comparing the detail coefficients with a given threshold value, and **shrinking these coefficients close to zero to take away the effect of noise** in the data [8]. The image is reconstructed from the modified coefficients. This process is known as the **inverse discrete wavelet transform**. During thresholding, a wavelet coefficient is compared with a given threshold and is set to zero if its magnitude is less than the threshold; otherwise, it is retained or modified depending on the threshold rule [17]. Thresholding distinguishes between the coefficients due to noise and the ones consisting of important signal information [14].

The choice of a threshold is an important point of interest. It plays a major role in the removal of noise in images because denoising will often produce smoothed images thereby reducing the sharpness of the image. Care must be taken to preserve the edges of the denoised image [21]. The various methods for

wavelet thresholding rely on the **choice of the threshold value**. Listed in this report are three well known methods for image noise removal [21]:

VisuShrink

SureShrink

BayesShrink.

There are two general categories of thresholding. They are hard- thresholding and soft-thresholding types. The hard-thresholding TH can be defined as [14]:

$$T_H = \begin{cases} x & \text{for } |x| \geq t \\ 0 & \text{in all regions} \end{cases} \quad (3.10)$$

Here t is the threshold value. A plot of T_H is shown in Figure 3.6.

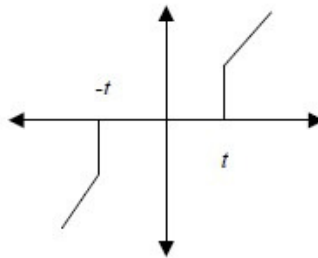


Figure 3.6: Hard thresholding

Thus, all coefficients whose magnitude is greater than the selected threshold value t remain as they are and the others with magnitudes smaller than t are set to zero. It creates a region around zero where the coefficients are considered negligible.

Soft thresholding is where the coefficients with greater than the threshold are shrunk towards zero after comparing them to a threshold value. It is defined as follows [16]:

$$T_H = \begin{cases} \text{sign}(x)(|x| - t) & \text{for } |x| \geq t \\ 0 & \text{in all regions} \end{cases} \quad (3.11)$$

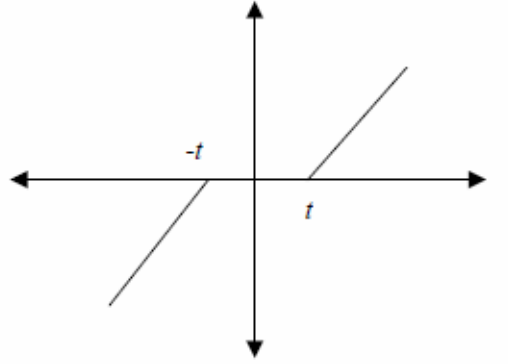


Figure 3.7: Soft thresholding

The soft yields more visually pleasant images [14]. The hard method is discontinuous and yields abrupt artifacts in the recovered images. The soft method yields a smaller minimum mean squared error compared to hard form of thresholding [14].

3.5.1 VisuShrink

VisuShrink was introduced by Donoho [16]. It uses a threshold value t that is proportional to the standard deviation of the noise. It follows the hard thresholding rule. It is also referred to as universal threshold and is defined as:

$$t = \sigma\sqrt{2 \log n} \quad (3.12)$$

σ^2 is the noise variance present in the signal and n represents the signal size or number of samples. An estimate of the noise level σ was defined based on the median absolute deviation given by:

$$\hat{\sigma} = \frac{\text{median}(\{|g_{i-j,k}| : k = 0, 1, \dots, 2^{j-1} - 1\})}{0.6745} \quad (3.13)$$

where $g_{j-1,k}$ corresponds to the detail coefficients in the wavelet transform.

VisuShrink does not deal with minimizing the mean squared error [37]. It is a general-purpose threshold selector that exhibit optimal minimax error properties and ensures that the estimates are as smooth as the true underlying functions. VisuShrink is known to yield recovered images that are overly smoothed [1]. This is because VisuShrink removes too many coefficients [8]. A disadvantage is that it does not remove speckle noise. It can only deal with an additive noise. VisuShrink follows the global thresholding scheme where there is a single value of threshold applied globally to all the wavelet coefficients.

The VisuShrink algorithm has been implemented using Matlab 6.5 [31].

3.5.2 SureShrink

A threshold chooser based on Stein's Unbiased Risk Estimator (SURE) was proposed by Donoho and Johnstone and is called SureShrink [30]. It is a combination of the universal threshold and the SURE threshold. This method specifies a threshold value t_j for each resolution level j in the wavelet transform

which is referred to as level dependent thresholding [1]. The goal of SureShrink is to minimize the mean squared error, defined as [6]:

$$\text{MSE} = \frac{1}{n^2} \sum_{x,y=1}^n (z(x,y) - s(x,y))^2 \quad (3.14)$$

where $z(x,y)$ is the estimate of the signal while $s(x,y)$ is the original signal without noise and n is the size of the signal. SureShrink suppresses noise by thresholding the empirical wavelet coefficients [30]. The SureShrink threshold t^* is defined as:

$$t^* = \min(t, \sigma\sqrt{2\log n}), \quad (3.15)$$

where t denotes the value that minimizes Stein's Unbiased Risk Estimator, σ is the noise variance computed from Equation (3.15), and n is the size of the image. SureShrink follows the soft thresholding rule. The thresholding employed is adaptive, i.e., a threshold level is assigned to each dyadic resolution level by the principle of minimizing the Stein's Unbiased Risk Estimator for threshold estimates [1]. It is smoothness adaptive, which means that if the unknown function contains abrupt changes or boundaries in the image, the reconstructed image also does.

3.5.3 BayesShrink

BayesShrink was proposed by Chang, Yu and Vetterli [6]. The goal of this method is to minimize the Bayesian risk, and hence its name, BayesShrink [44]. It uses soft thresholding and is subband-dependent, which means that thresholding is done at each band of resolution in the wavelet decomposition. Like the SureShrink procedure, it is smoothness adaptive.

The Bayes threshold, t_B , is defined as:

$$t_B = \frac{\sigma^2}{\sigma_s^2} \quad (3.16)$$

where σ^2 is the noise variance and σ_s^2 is the signal variance without noise. The noise variance σ^2 is estimated from the subband HH1 in Figure 3.3 by the median estimator shown in Equation (3.16). From the definition of additive noise we have:

$$w(x, y) = s(x, y) + n(x, y). \quad (3.17)$$

Since the noise and the signal are independent of each other, it can be state that:

$$\sigma_w^2 = \sigma_s^2 + \sigma^2 \quad (3.18)$$

σ_w^2 can be computed as shown below:

$$\sigma_w^2 = \frac{1}{n^2} \sum_{x,y=1}^n (w^2(x, y)) \quad (3.19)$$

The variance of the signal σ_s^2 is computed as:

$$\sigma_s = \sqrt{\max(\sigma_w^2 - \sigma^2, 0)} \quad (3.20)$$

with σ^2 and σ_s^2 , the Bayes threshold is computed from Equation (3.16). Using this threshold, the wavelet coefficients are thresholded at each band shown in Figure 3.5.

3.6 Wavelet Filters in MATLAB

There are fifteen families of Wavelet filters available in the MATLAB software [31]. The families of Wavelet filters are displayed in Figures 3.8, 3.9, and 3.10.

```

Haar                                haar
=====
Daubechies                          db
-----
db1 db2 db3 db4
db5 db6 db7 db8
db9 db10 db**
=====
Symlets                              sym
-----
sym2 sym3 sym4 sym5
sym6 sym7 sym8 sym**
=====
Coiflets                             coif
-----
coif1 coif2 coif3 coif4
coif5
=====
BiorSplines                         bior
-----
bior1.1 bior1.3 bior1.5 bior2.2
bior2.4 bior2.6 bior2.8 bior3.1
bior3.3 bior3.5 bior3.7 bior3.9
bior4.4 bior5.5 bior6.8
=====

```

Figure 3.8 Wavelet Families
Haar, Daubechies, Symlets
Coiflets
BiorSplines

```

=====
ReverseBior                         rbio
-----
rbio1.1 rbio1.3 rbio1.5 rbio2.2
rbio2.4 rbio2.6 rbio2.8 rbio3.1
rbio3.3 rbio3.5 rbio3.7 rbio3.9
rbio4.4 rbio5.5 rbio6.8
=====
Meyer                              meyr
=====
DMeyer                             dmey
=====
Gaussian                           gaus
-----
gaus1 gaus2 gaus3 gaus4
gaus5 gaus6 gaus7 gaus8
gaus**
=====
Mexican_hat                         mexh
=====
Morlet                             morl
=====
Complex Gaussian                   cgau
-----
cgau1 cgau2 cgau3 cgau4
cgau5 cgau**
=====

```

Figure 3.9 Wavelet Families
ReverseBior, Myer, DMeyer,
Gaussian, Mexican_hat
Morlet, Complex Gaussian

```

=====
Shannon                shan
-----
shan1-1.5  shan1-1  shan1-0.5  shan1-0.1
shan2-3  shan**
=====
Frequency B-Spline    fbsp
-----
fbsp1-1-1.5 fbsp1-1-1  fbsp1-1-0.5 fbsp2-1-1
fbsp2-1-0.5 fbsp2-1-0.1 fbsp**
=====
Complex Morlet        cmor
-----
cmor1-1.5  cmor1-1  cmor1-0.5  cmor1-1
cmor1-0.5  cmor1-0.1  cmor**
=====

```

Figure 3.10 Wavelet Families
Shannon, Frequency, Complex
Morlet

The different threshold selections provide the option for selecting the Wavelet Type [19].

3.7 Visushrink Example

Matlab 6.5 has a Wavelet Tool box for performing detailed wavelet transform analysis [31]. The following wavelet commands are important for wavelet processing:

- 1) `Imread ()`
- 2) `Imnoise ()`
- 3) `Ddencomp ()`
- 4) `Wdencomp ()`
- 5) `Imwrite ()`.

It is required to have a full understanding of the commands to effectively utilize the Wavelet solution. The VisuShrink algorithm has been implemented utilizing the version Matlab 6.5 [31]. Latest versions including R2014 and R2015 are available. For this report the Matlab 6.5 version is sufficient for most needs. However for other methods including the Sureshrink Color implementation, the R2014 and R2015 versions are required. The “Woman” image and the “Cameraman” images are read using the **imread ()** function. Noise is added to the image using **imnoise ()**. The threshold value using Equation (4.8) is computed from the function **ddencomp ()**. A global thresholding is applied over the wavelet coefficients with **wdencomp ()**. The modified image is saved with **imwrite ()** function. The images subjected to the VisuShrink thresholding are shown from Images 3.8 through 3.13. Biorthogonal type ‘**bior 6.8**’ wavelet was applied to the images. The denoised images are shown below. The Matlab 6.5 functions are given in Appendix A



Image 3.8: Cameraman



Image 3.9: Image Corrupted
with Gaussian Noise, Variance 0.01
Cameraman



Image 3.10: Image after Application
of VisuShrink
MSE=0.0370, MAE=0.0431, PSNR=34.31



Image 3.11 Original Image
Woman



Image 3.12: Noisy Image with
Random Noise Generation
RAND Function



Image 3.13: Denoised Image
application of Visushrink
PSNR=18.70

3.8 SureShrink Example

SureShrink filtering is available in version Matlab 6.5 [31]. With the function **imread ()**, the image “cameran.tif” is loaded into the Matlab workspace. This image is corrupted with Gaussian noise with the help of **imnoise ()** function. Daubechies wavelet decomposition is done on the corrupted image with **wavedec2 ()** function. The threshold values are computed using the function, **wdcbm2 ()**. Sure thresholding is done on the detail coefficients with **wdencmp ()** specifying the parameter, ‘**lvd**’ in the function which means that **level dependent thresholding** is done on the coefficients. Image 3.14 is the image corrupted with Gaussian noise (variance=0.05). Image 3.15 is the one obtained after Image 3.14 is subjected to SureShrink thresholding.



Image 3.14: Input corrupted
Gaussian Noise-Variance 0.05



Image 3.15: Image after SureShrink
MSE=0.2020, MAE=0.0568, PSNR=13.48

3.9 Bayes Example

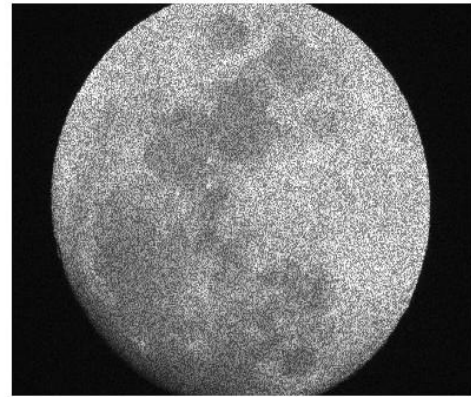
Matlab 6.5 [31] is used for the application of BayesShrink. The image “cameraman.tif” is loaded into the workspace by using **imread ()**. This image is corrupted with Gaussian noise using the **imnoise ()** function. The images obtained are subjected to a discrete wavelet transform using as examples both ‘**sym4**’ (**Symlets**) and ‘**db4**’ (**Daubechies**) wavelets with the help of the **dwt2 ()** function. This function generates wavelet coefficients for the corrupted image. There are four bands namely, cA, cH, cV and cD, where cA corresponds to the approximation coefficients, while cH, cV, and cD are the detail coefficients over which thresholding is done. The noise variance for each band is computed using Equation (3.9) and the signal variance is computed using Equation (4.11). With these two values, the threshold value is computed from Equation (4.10). Thresholding of the wavelet coefficients is brought about using the function **wthresh ()**. Inverse wavelet transform using **idwt2 ()** is done on the modified wavelet coefficients to get the signal. The image is built from this signal using **imwrite ()** function. In Image 3.19, BayesShrink has been applied (wavelet-db4) to remove speckle noise with variance of 0.06. In Image 3.23, the results of denoising (wavelet-sym4) an image with Gaussian noise (mean=0, variance = 0.05) is shown. In Image 3.27, the results of denoising (wavelet-bior 6.8) an image with Salt and Pepper noise is shown.

original image

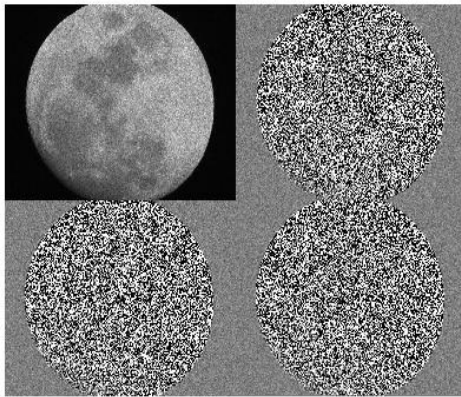


Image 3.16 Full Moon

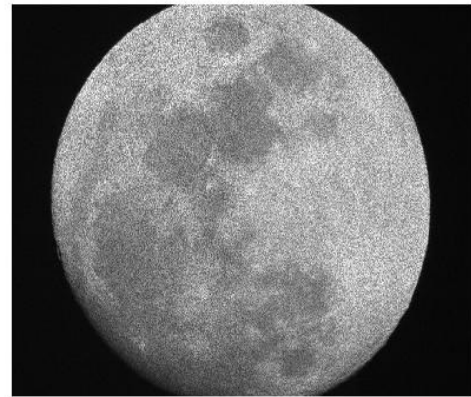
noisy image

Image 3.17 Noisy Image
Corrupted with Speckle Noise

decomposed structure

Image 3.18
Decomposed Image

denoised image

Image 3.19 Image Denoised
with wavelet db4
Variance 0.06

Performance Evaluation Metrics For Wavelet db4	
PSNR	33.56
MSE	0..085
Variance	0.06

Table 3.1 Performance Evaluation Metrics Wavelet db4

original image

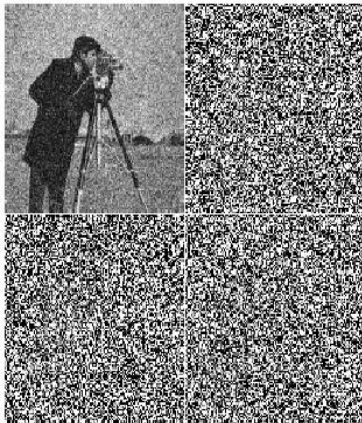


Image 3.20 Cameraman

noisy image

Image 3.21 Noisy Image
Corrupted with Gaussian Noise

decomposed structure

Image3.22
Decomposed Image

denoised image

Image 3.23 Image Denoised
with wavelet sym4
Variance 0.05

Performance Evaluation Metrics For Wavelet sym4	
PSNR	27.93
MSE	0.1208
Variance	0.06

Table 3.2 Performance Evaluation Metrics Wavelet sym4

original image



Image 3.24 Woman

noisy image



Image 3.25 Noisy Image Corrupted with Salt and Pepper Noise

decomposed structure

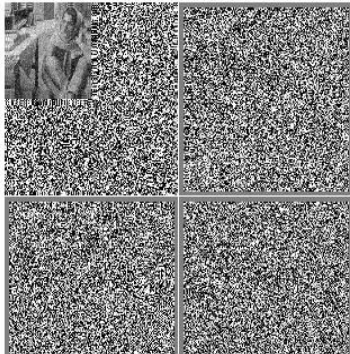


Image 3.26 Decomposed Image

denoised image



Image 3.27 Image Denoised with wavelet bior 6.8

Performance Evaluation Metrics For Wavelet bior 6.8	
PSNR	35.06
MSE	0.1278
MAE	0.0873

Table 3.3 Performance Evaluation Metrics Wavelet bior 6.8

3.10 Summary

Denoising of images using VisuShrink, SureShrink and BayesShrink using Matlab 6.5 is discussed in this chapter. All these methods are based on the application of wavelet transforms. Each of these methods is compared in terms of the PSNR, the MSE, and the MAE error criteria discussed in Chapter 1 of this thesis.

Chapter 4

4.0 Image Processing Applications in Science and Medicine

4.1 Astronomy

There are numerous areas of application utilizing digital image processing techniques. The principal energy source for images in use today is the electromagnetic energy spectrum. Example images utilized in the study of astronomy are presented below. Images of the Crab Pulsar are displayed in five electromagnetic spectrums including Gamma, X-ray, Optical, Infrared, and Radio. Images of the Crab Pulsar (five different spectral bands) are shown [18] (Images 4.1-4.5).

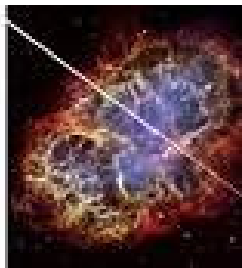


Image 4.1 Gamma

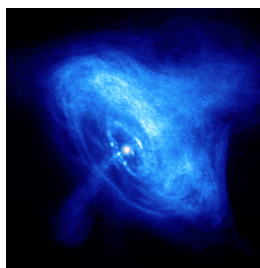


Image 4.2 X-ray



Image 4.3 Optical

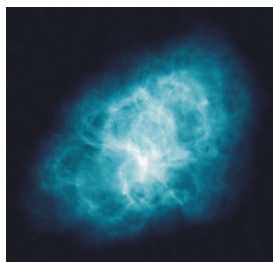


Image 4.4 Infrared



Image 4.5 Radio

The advances in digital camera optics have advanced the studies of Astronomy greatly. Consumer cameras can now give quality pictures with mega-pixels in the 18-24 range at reasonable cost. The following pictures are the courtesy of professional photographer (Shawn Simone: Woodland Hills, California). These photographs were taken with a Nikon D3200 (\$500.00 camera).



Figure 4.1 Full Moon
8x Zoom Mar 2016



Figure 4.2 Full Moon
5x Zoom Mar 2016



Figure 4.3 Full Moon
10x Zoom Mar 2016

4.2 Medicine

4.2.1 Tomographic Imaging

Major applications of Tomographic imaging in medicine include CT scans, MRI imaging, X-ray imaging, Acoustic imaging, Ultrasonic detection, and electron microscopy [2]. Tomography is a form of X-ray imaging that attenuates the effects of objects that lie in front of or behind a perpendicular plane of interest. It creates a cross-sections image of a solid object. The technique is implemented by moving an X-ray radiation source and a piece of film in parallel and opposing directions, pivoting about the plane of interest [2]. The information along the plane is constructively reinforced on the film, while information outside the plane is not. Computed Tomography combines the series of X-ray images taken from different angles using a computer to create cross-sectional images of the bones, blood vessels and soft tissues inside your body. The resulting image is a cross-section of the object about the pivot plane. CT scan images provide more detailed information than plain X-rays do. A CT scan can be used to visualize nearly all parts of the body and is used to diagnose disease or injury as well as to plan medical, surgical or radiation treatment [33].

Computed tomography is used in three primary modes, Transmissive, Emissive, and Reflective.

4.2.1.1 Transmissive Mode

The original technique is the transmissive mode and uses an X-ray radiation source [2]. X-rays are transmitted through the imaged object and received at X-ray detection devices. This classical form of computed tomography is referred to simply as CT [18]. Ultrasonic pressure sources and associated detectors can also be used to create transmissive computed tomography images in the same way.

4.2.1.2 Emissive Mode

The emissive mode of computed tomography relies on the emission of a detectable signal from the imaged object [41]. The object can be directly excited or a substance can be introduced that is excited. Detectors receive the emitted signal. Two emissive mode systems are common, **Magnetic Resonance Imaging** (MRI) and **Positron Emission Tomography** (PET). Radio waves are used in Magnetic Resonance Imaging [2]. The technique places a patient in a powerful magnet and passes radio waves through the body in short pulses. Each pulse causes a responding pulse of radio waves to be emitted by the patient's tissues. The location from which these signals originate and the strength of the signals is determined by a computer. This produces a two-dimensional picture of a section of the patient. The MRI will produce pictures in several planes. PET imaging introduces a positron-emitting substance into the imaged object. The substance emits a constant flow of positrons. As the emitted positrons come to rest, they interact with an electron and create two gamma-ray photons, each traveling away from the other. Two detectors track the coincidental photons and

determine the location of the positron emission. The term angiography is based on projectional radiography and the term has been applied to newer vascular imaging techniques such as CT angiography and MR angiography [2].

4.2.1.3 Reflective Mode

The third mode of computed tomography is the reflective mode [2]. As in the transmissive mode, a source transmits a signal at the imaged object. Instead of passing through the object, the signal enters the object and is reflected by the internal elements of the object back out to a detector device. The received signal at the detector is proportional to the density of the elements of the imaged object. Reflective computed tomography has the advantage of not requiring the imaged object to be surrounded with sources and detectors.

Computed Tomography Images are shown in Image 4.4 thru Image 4.23. Twenty four slices were takes in this MRI scan. Twenty of the pictures are being displayed. The following Medical scans are the courtesy of the Tarzana Medical Center in Woodland Hills, California.



Figure 4.4 Ax T1 SE slice #1

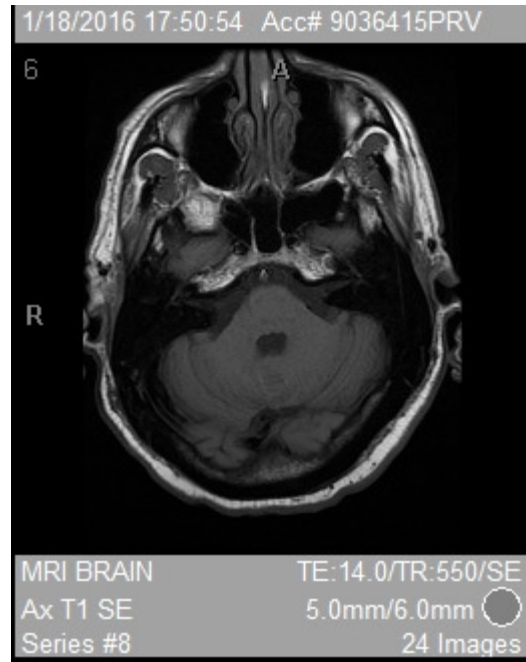


Figure 4.5 Ax T1 SE slice #6



Figure 4.6 Ax T1 SE slice #12



Figure 4.7 Ax T1 SE slice #14

MRI Angiogram images of the Carotid are displayed.

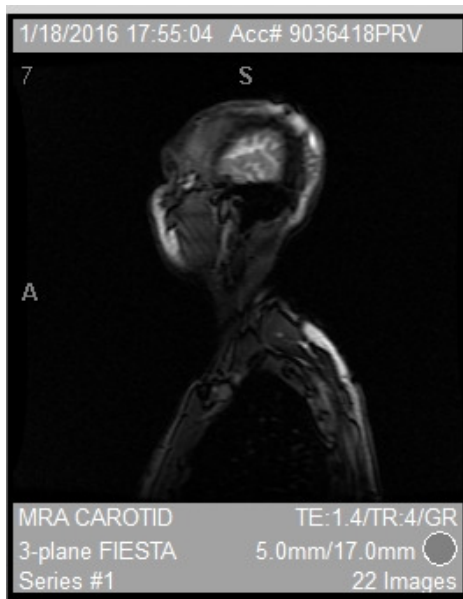


Figure 4.8 MRA Carotid/3-Plane Fiesta S7

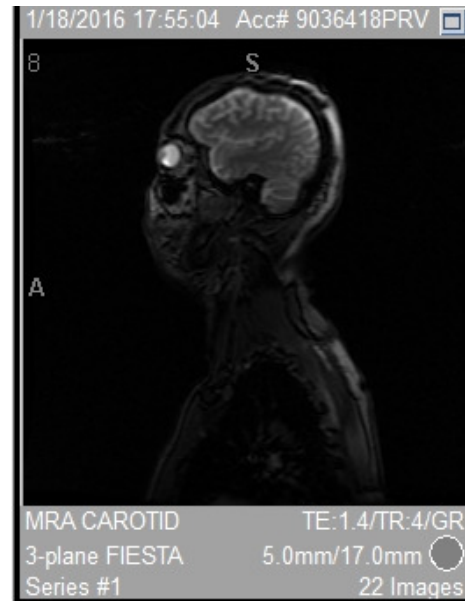


Figure 4.9 MRA Carotid/3-Plane Fiesta S8

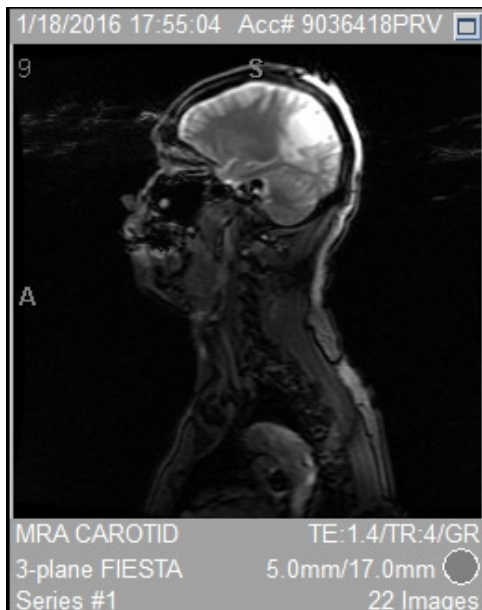


Figure 4.10 MRA Carotid/3-Plane Fiesta S9

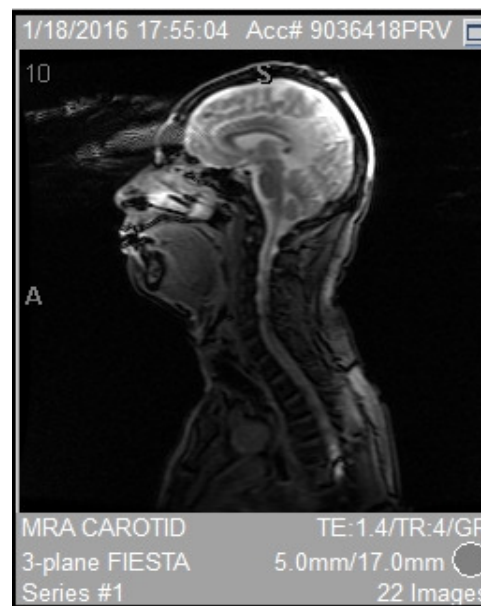


Figure 4.11 MRA Carotid/3-Plane Fiesta S10

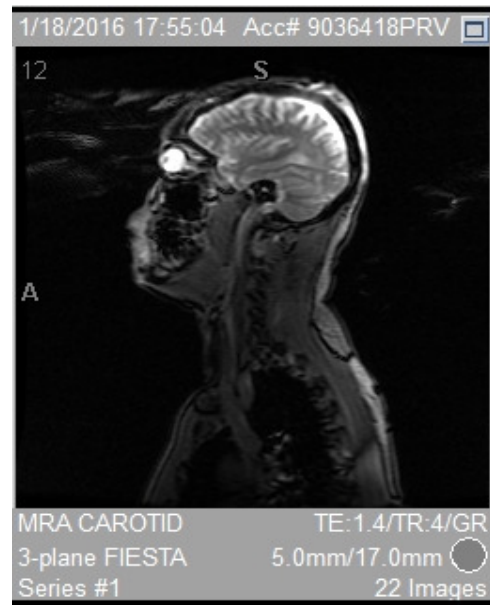
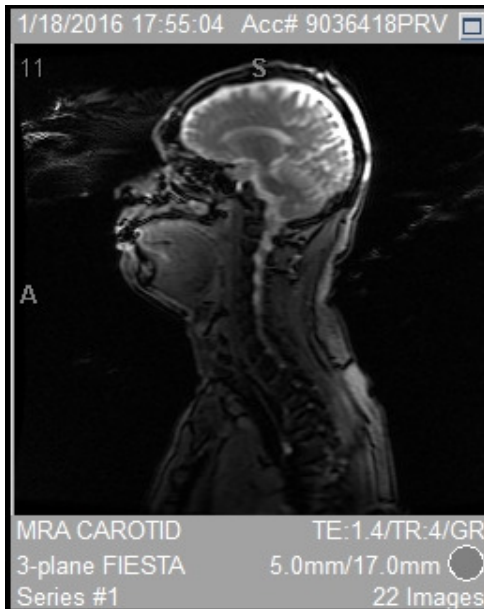


Figure 4.12 MRA Carotid/3-Plane Fiesta S11 Figure 4.13 MRA Carotid/3-Plane Fiesta S12

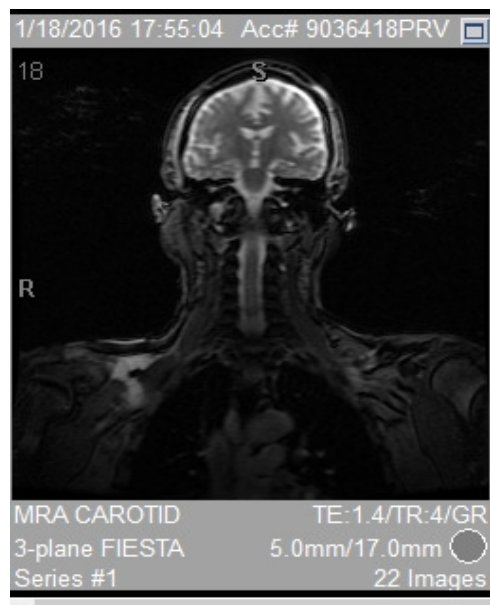
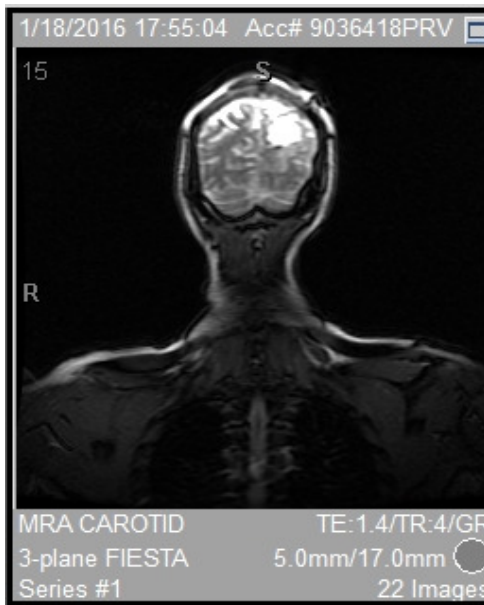


Figure 4.14 MRA Carotid/3-Plane Fiesta S15 Figure 4.15 MRA Carotid/3-Plane Fiesta S18

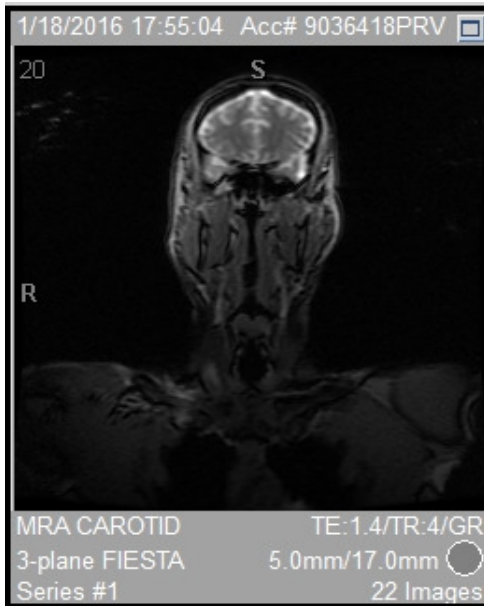


Figure 4.16 MRA Carotid/3-Plane Fiesta S15 Figure 4.17 CT Head WO Contrast Slice 1

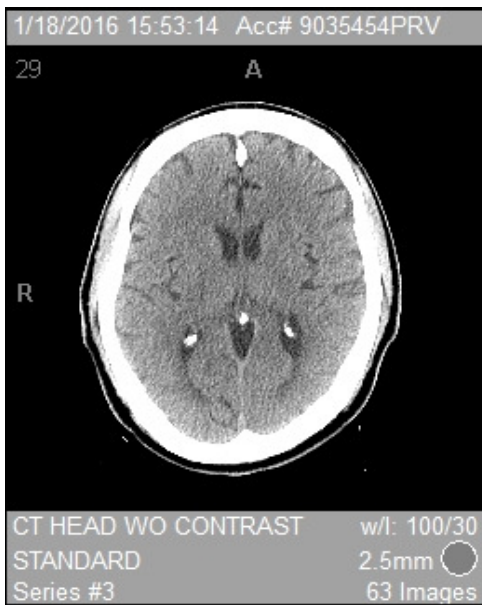


Figure 4.18 CT Head WO Contrast Slice 29 Figure 4.19 CT Head WO Scout Slice 1

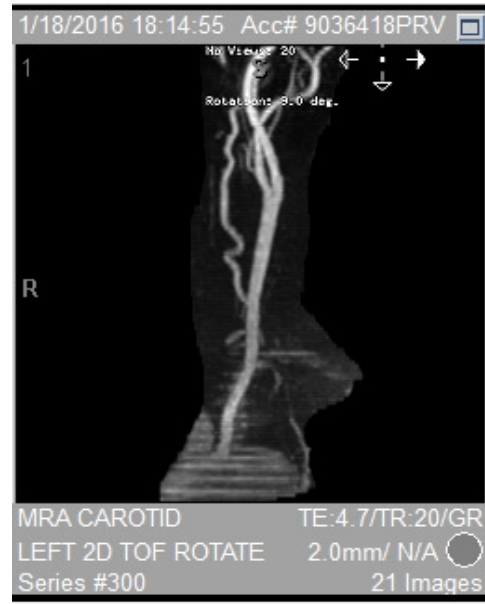


Figure 4.20 CT Head WO Scout Slice 2 Figure 4.21 MRA Carotid Left 2D TOF Rotate S1

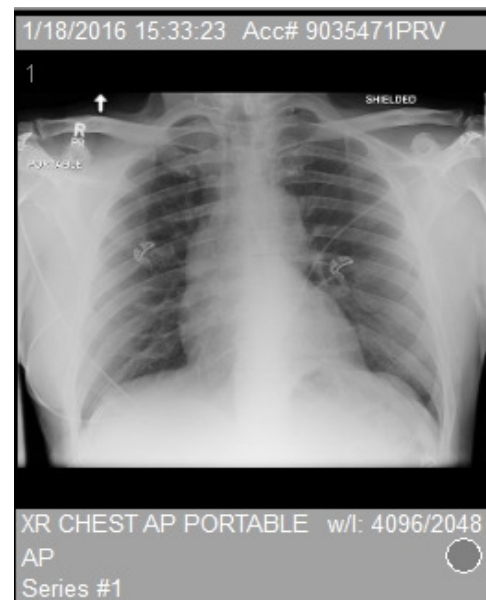


Figure 4.22 MRA Carotid Left 2D TOF Rotate S11 Figure 4.23 XR Chest AP Portable

4.3 Experiment Results

4.3.1 Performance Evaluation - Denoising Technique

To get the measure of the wavelet filter performance, experimental results are evaluated according to three error criteria: the mean square error (MSE), the mean absolute error (MAE) and peak signal to noise ratio (PSNR). These three parameters decide which wavelet and thresholding technique gives the best result. Definitions of the Assessment Criteria are shown below in Figure 4.24 [24].

Assessment	Description	Formula
SNR	Signal-to-noise ratio expressed in dB	$SNR = 10 \cdot \log_{10} \left[\frac{\sum_{x=0}^{n_x-1} \sum_{y=0}^{n_y-1} [r(x, y)]^2}{\sum_{x=0}^{n_x-1} \sum_{y=0}^{n_y-1} [r(x, y) - t(x, y)]^2} \right]$
PSNR	Peak signal-to-noise ratio expressed in dB	$PSNR = 10 \cdot \log_{10} \left[\frac{\max(r(x, y))^2}{\frac{1}{n_x \cdot n_y} \cdot \sum_{x=0}^{n_x-1} \sum_{y=0}^{n_y-1} [r(x, y) - t(x, y)]^2} \right]$
RMSE	Root mean square error	$RMSE = \sqrt{\frac{1}{n_x n_y} \cdot \sum_{x=0}^{n_x-1} \sum_{y=0}^{n_y-1} [r(x, y) - t(x, y)]^2}$
MAE	Mean absolute error	$MAE = \frac{1}{n_x n_y} \cdot \sum_{x=0}^{n_x-1} \sum_{y=0}^{n_y-1} r(x, y) - t(x, y) $

Figure 4.24

(Reference image $r(x, y)$, Test image $t(x, y)$, Size $[n_x, n_y]$)

A high PSNR means good quality and low means bad quality. Note that the PSNR calculation uses the term mean square error (MSE) in the denominator. So, if the MSE error is low then the PSNR will be high. There is an inverse relationship between PSNR and MSE.

Bear in mind that none of these objective measures are particularly good at predicting human visual response to image quality. Sometimes PSNRs vary wildly between two almost indistinguishable images; similarly you can have two images with the same PSNR where there is a very obvious difference in quality. There are a number of reasons why MSE or PSNR may not correlate well with the human perception of quality [20]. Digital pixel values, on which the MSE is typically computed, may not exactly represent the light stimulus entering the eye. Simple error summation, like the one implemented in the MSE formulation, may be markedly different from the way the Human Visual System and the brain arrives at an assessment of the perceived distortion. Two distorted image signals with the same amount of error energy may have very different structure of errors, and hence different perceptual quality [20].

There are other measurements which made be applied to determine image quality. SSIM Index (structural similarity index measurement) is based on measuring the three components (luminance similarity, contrast similarity and structural similarity) and combining them into a resultant value. This metric is not reviewed in this report.

Note, that a 20 dB or higher PSNR is desired for an image to be consider an image of good quality [18].

Additive noise with a uniform distribution was added to a MRI (Image 4.6) image in order to corrupt the real MRI test image and thus access the performance of the noise filter algorithms.

An example of the original Image, Noisy Image, and Denoised Image are shown in Image 4.6, Image 4.7, and Image 4.8, respectively.

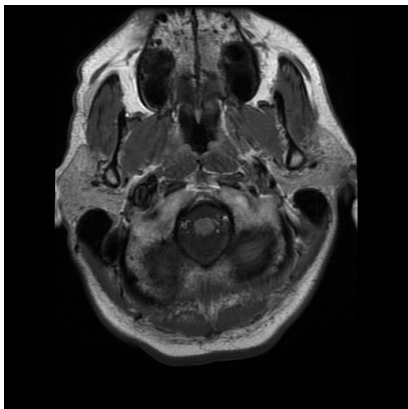


Image 4.6 Original MRI Image

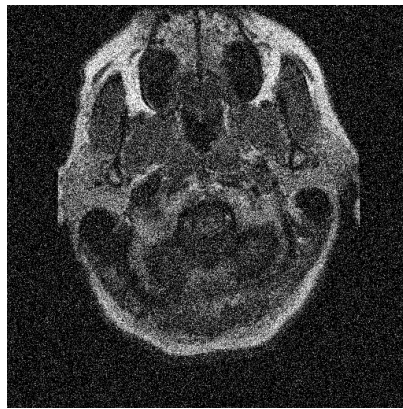


Image 4.7 Noisy MRI Image

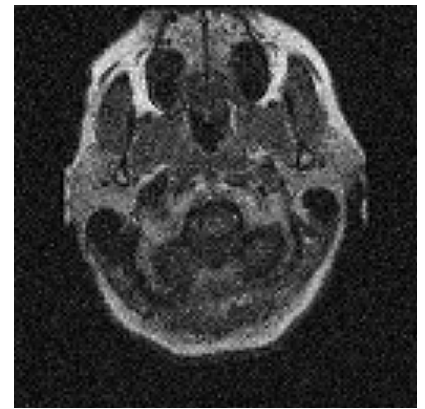


Image 4.8 Denoised MRI Image

The procedures taken for the performance evaluation include the following steps:

- a) Add additive uniform noise to image
- b) Select Wavelet Type
- c) Select level of decomposition
- d) Apply Thresholding technique
- e) De-noise the image

- f) Compute the Mean Square Error (MSE)
- g) Compute the Mean Absolute Error (MAE)
- h) Compute the Peak Signal to Noise-Ratio (PSNR).

Threshold plays an important role in the denoising process. A small threshold value will retain the noisy coefficients whereas a large threshold value leads to the loss of coefficients that carry image signal details. Hard thresholding and soft thresholding techniques are used for de-noising process. Hard thresholding is a keep or kill rule whereas soft thresholding shrinks the coefficients above the threshold in absolute.

There are a variety of ways to determine the threshold value T . Depending on whether or not the threshold value T changes across wavelet scales and spatial locations the thresholding can be:

- 1) Global Thresholding
- 2) Level Dependent Thresholding
- 3) Optimal Thresholding.

1. Global Threshold: a single value T is applied globally to all empirical wavelet coefficients at different scales. $T = \text{constant}$.
2. Level-Dependent Threshold: a different threshold value T is selected for each wavelet analysis level (scale). $T = T(j)$, $j=1, 2, \dots, J$. J is the coarsest level for wavelet expansion to be processed.

3. Optimal Threshold: Estimate the mean square error function so that the error of the output to minimize the function, the minimum MSE serves as a solution to the optimal threshold.

The MATLAB wavelet tool box is utilized to implement the Global Threshold selection. The following command with option selection for **Global** is chosen:

- a) `WDENCMP ('gbl', X, 'wname', N, THR, SORH, KEEPAPP)`
returns a de-noised version of input signal X (2-D) obtained by wavelet coefficients thresholding using **Global** Positive Threshold THR
- b) Wavelet decomposition is performed **at level N** and
'wname' is a string containing the **wavelet name**.
SORH ('s' or 'h') is for **soft or hard thresholding**. If KEEPAPP = 1, approximation coefficients cannot be thresholded, otherwise it is possible.

For Level-Dependent Threshold selection the following command is executed:

- a) `WDCBM2` Thresholds for wavelet 2-D using Birge-Massart strategy.
`[THR,NKEEP] = WDCBM2(C, S, ALPHA, M)` returns a **Level-dependent** thresholds THR and numbers of coefficients to be kept NKEEP, for de-noising. THR is obtained using a wavelet coefficients selection rule based on Birge-Massart strategy.

Figure 4.25 is an illustration of the improvement in image quality as the PSNR value increases.

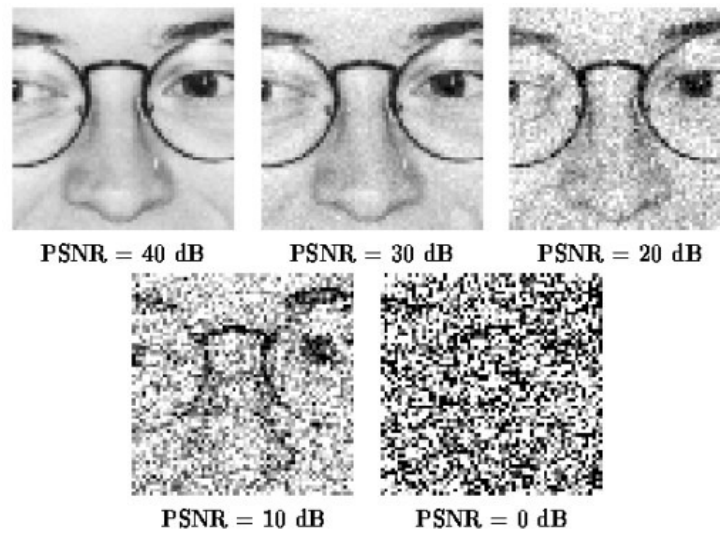


Figure 4.25

Illustration of Image Improvement for increased PSNR measurement

The image with PSNR value of 40 db shows a large improvement in quality compare to the image with a PSNR value of 10 db. The PSNR calculation appears to be a reasonable parameter in assessing the quality of a denoised digital image.

4.3.2 Table of Results Comparison

For comparison of the eight different wavelet functions, the quantitative de-noising results of the MRI images (Image 4.8) obtained by using global dependent and level dependent thresholding are shown in Tables 4.1 and 4.2 (Appendix A – denoise.m). The MSE, MAE, and PSNR error criteria are the ones which have been used to assess the performance of the wavelet functions. The numerical results are summarized in the following tables.

Table 4.1 Quality Analyses (MRI Image) -GLOBAL THRESHOLDING

Type of Wavelet	Level 1		
	MSE	MAE	PSNR (db)
Haar	0.1635	0.0725	14.8163
db2	0.1627	0.0697	16.2429
db4	0.1624	0.0694	15.6793
sym2	0.1628	0.0700	16.2924
sym4	0.1622	0.0687	15.7321
bior1.1	0.1646	0.0731	14.8739
bior1.3	0.1694	0.0733	15.0875
bior6.8	0.1626	0.0694	15.7582
dmey	0.1622	0.0690	15.5582
coif5	0.1622	0.0689	15.5413

Table 4.2 Quality Analyses (MRI Image) - LEVEL DEPENDENT
THRESHOLDING

Type of Wavelet	Level 1		
	MSE	MAE	PSNR (db)
Haar	0.2081	0.0633	11.7748
db2	0.2062	0.0593	13.4776
db4	0.2041	0.0581	13.4637
sym2	0.1628	0.0700	13.4139
sym4	0.2051	0.0579	13.7844
bior1.1	0.2076	0.0609	12.0776
bior1.3	0.2077	0.0606	12.5512
bior6.8	0.2052	0.0582	13.1794
dmey	0.2019	0.0573	13.6780
coif5	0.2028	0.0579	13.5676

To improve the quality of an image, the value of the MSE should be as low as possible while the value of PSNR should be higher. From the above Tables 4.1 and Table 4.2, analysis showed that the **symlet2** and **symlet4** mother wavelets provide a low mean square error and low mean absolute error. In addition the PSNR value is higher as compare to other wavelet choices. In this study, the symlet wavelet can be considered as the most suitable mother function when processing signal denoising. Also, it is clear from Table 4.1 and Table 4.2 that there is an improvement in the quality (low MSE and high PSNR) of the image after de-noising and adjustment.

4.4 Generalization into other Scenarios

3D MRI Imaging

Although Computer Tomography (CT) scans, Magnetic Resonance Imaging (MRI) scans, and Astronomical Data Visualizations are scientific fields fundamentally based on imaging, the visualization tools commonly used are for **2D** imaging despite the existence of **3D** data sets. The development of 3D visualization tools in Astronomy and Medical Imaging has only begun over the past few years, and these tools generally lack the usability and the up-to-date graphic capabilities since they are based on astronomy or medical toolkits developed for **2D visualization**.

Harvard University has initiated a project, The Astronomical Medicine Project [4] at the Initiative in Innovative Computing Center, to extend the understanding of complex data sets in the fields of medical imaging and astronomy using a broad-based approach to data exploration and analysis. In the same way astronomers want to be able to visualize data from a radio telescope in 3D and identify young stars; physicians want to be able to visualize a MRI scan in 3D and have the capability of segmenting out a tumor. By combining the expertise of medical imaging and astronomy through shared software and professional collaboration, the Astronomical Medicine project is developing tools and techniques to address the commonalities of research in the imaging sciences [4] .

4.4.1 Create Head Phantom – Two Dimensions

The MATLAB Tool kit has the ability to execute the command: PHANTOM; to generate a head phantom image. The test image is the Shepp-Logan head phantom which can be generated using the function phantom. The phantom image illustrates many qualities that are found in real-world tomographic imaging of human heads [31]. As a starting point, a two dimensional representation of the bright elliptical shell along the exterior is analogous to a skull and the many ellipses inside are analogous to brain features or tumors.

The following MATLAB command sequence will produce study models utilizing the head phantom image (see below):

```
P = phantom (128);
R = radon (P,0:179);
I1 = iradon(R,0:179);
I2 = iradon(R,0:179,'linear','none');
subplot (1,3,1), imshow (P), title ('Original')
subplot (1,3,2), imshow (I1), title ('Filtered backprojection')
subplot (1,3,3), imshow (I2, []), title ('Unfiltered backprojection')
```

The phantom plot geometry is displayed demonstrating the variations in back plane projection:

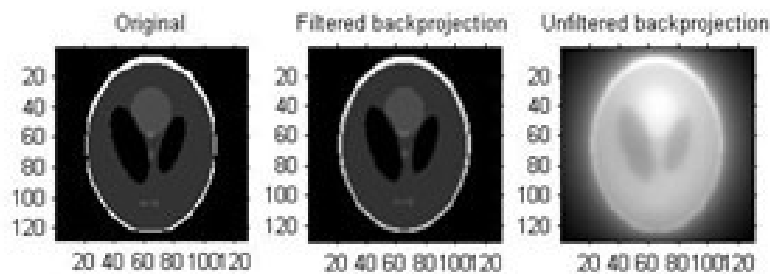


Figure 4.26 Phantom Geometry

4.4.2 Extension to Creating 3D MRI Images

The following artist conceptions and images are examples of how three-dimensional images would be used to enhance the overall perception of 3D MRI scans. The examples demonstrate the 3D reconstruction of the human brain [27].

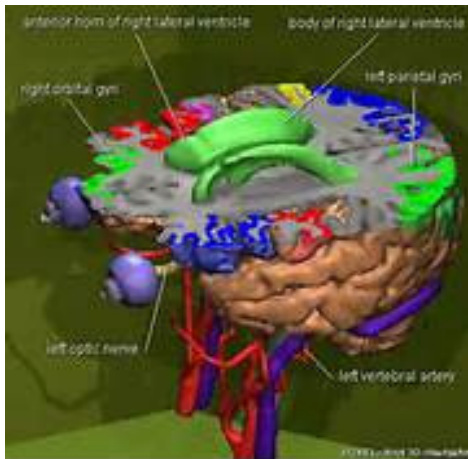


Figure 4.27 [9]

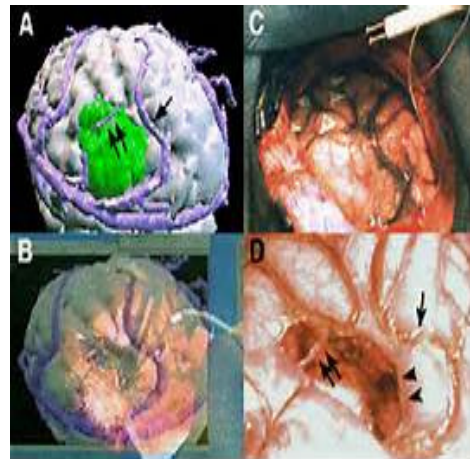


Figure 4.28 [10]

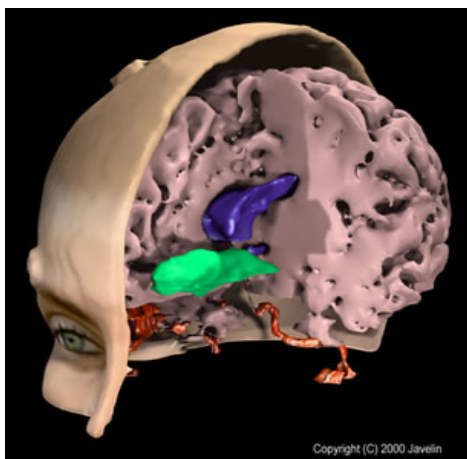


Figure 4.29 [11]

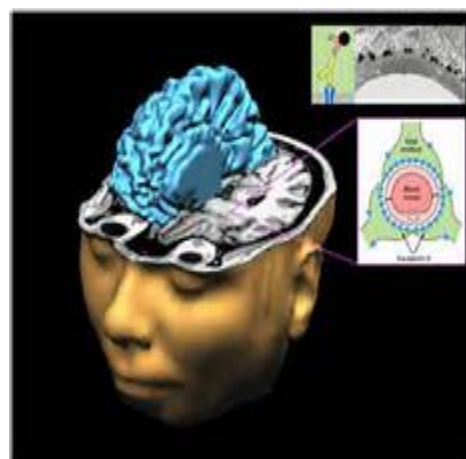


Figure 4.30 [12]

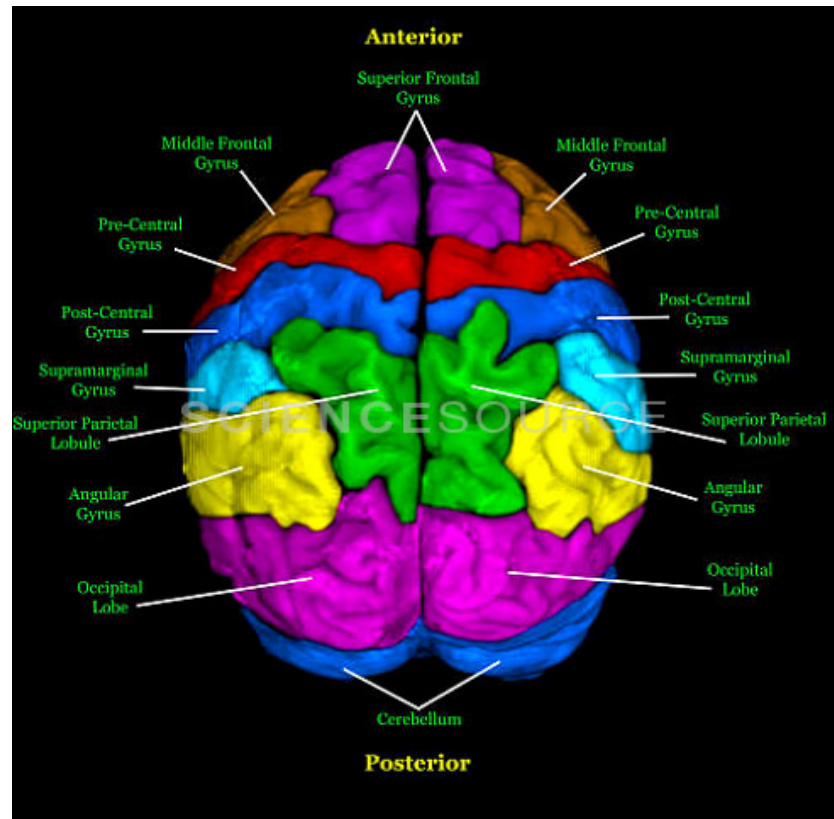


Figure 4.31 [13]

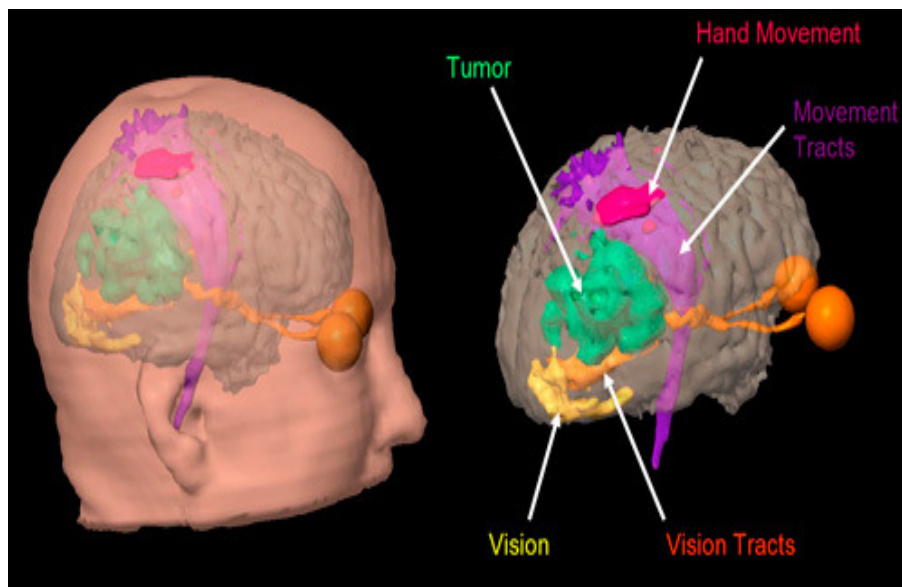


Figure 4.32 [13]

By applying 3D visualization techniques developed in both the Medical Science Discipline and the Astronomy Discipline, scientists will improve the way medical scientists are able to study and visualize 3D data sets [4]. An example of applying 3D visualization techniques within the MATLAB Tool kit is demonstrated with the following MATLAB example [31]. The MATLAB sequence shows how to analyze 3D data sets using a three-dimensional wavelet analysis tool, and how to display low-pass and high-pass components along a given slice. The example focuses on MRI (magnetic resonance images). A 3D plot of the Original Data and a 3D plot of the Level 2 Approximation are shown.

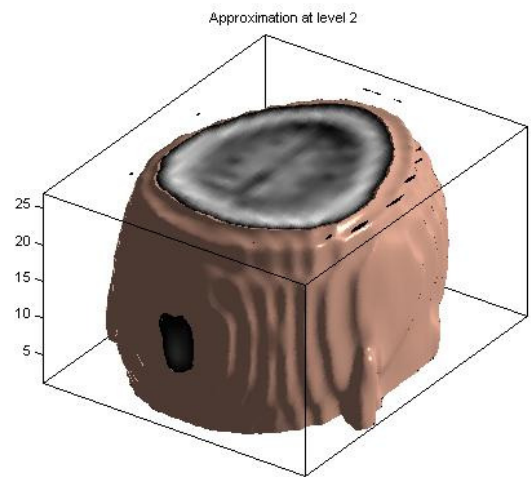
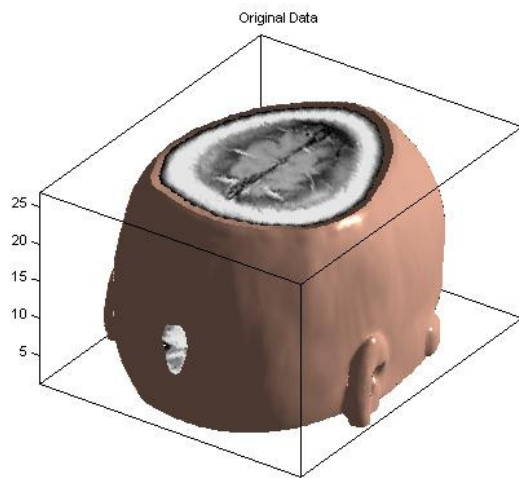


Figure 4.33 Original Data [31]

Figure 4.34 Level 2 Approximation [31]

The Matlab command options for the 3D MRI are summarized below.

Part 1:

```
% Wavelet Analysis for 3D Data
load wmr1
clear XP

% Multilevel 3D Wavelet Decomposition
% Compute the wavelet decomposition of the 3D data at level 3.
n = 3;           % Decomposition Level
w = 'sym4';      % Near symmetric wavelet
WT = wavedec3(X,n,w); % Multilevel 3D wavelet decomposition.

%% Multilevel 3D Wavelet Reconstruction
% Reconstruct from coefficients the approximations and details for levels 1 to 3.
A = cell(1,n);
D = cell(1,n);
for k = 1:n
    A[k] = waverec3(WT,'a',k); % Approximations (low-pass components)
    D[k] = waverec3(WT,'d',k); % Details (high-pass components)
end
%%
% Check for perfect reconstruction.
err = zeros(1,n);
for k = 1:n
    E = double(X)-A[k]-D[k];
    err(k) = max(abs(E(:)));
end
disp(err)
%% 3D Display of Original Data and Approximation at Level 2
% The size of the 3D original array X is "(128 x 128 x 27) = 442368". We
% can use a 3D display to show it.
figure('DefaultAxesXTick',[1,'DefaultAxesYTick',[1,...
    'DefaultAxesFontSize',8,'Color','w'])

XR = X;
Ds = smooth3(XR);
hiso = patch(isosurface(Ds,5),'FaceColor',[1,75,65],'EdgeColor','none');
hcap = patch(isocaps(XR,5),'FaceColor','interp','EdgeColor','none');
colormap(map)
daspect(gca,[1,1,4])
lightangle(305,30);
set(gcf,'Renderer','zbuffer'); lighting phong
isonormals(Ds,hiso)
```

Figure 4.35 MATLAB Code

Part 2:

```

set(hcap,'AmbientStrength',6)
set(hiso,'SpecularColorReflectance',0,'SpecularExponent',50)
set(gca,'View',[215,30],'Box','On');
axis tight
title('Original Data')
%%
% The 3D array of the coefficients of approximation at level 2, whose size
% is  $(22 \times 22 \times 9) = 4356$ , is less than 1% the size of the original data.
% With these coefficients, we can reconstruct A2, the approximation at
% level 2, which is a kind of compression of the original 3D array.
% A2 can also be shown using a 3D display.
figure('DefaultAxesXTick',[1,'DefaultAxesYTick',[1,...
'DefaultAxesFontSize',8,'Color','w')
XR = A[2];
Ds = smooth3(XR);
hiso = patch(isosurface(Ds,5),'FaceColor',[1,75,65],'EdgeColor','none');
hcap = patch(isocaps(XR,5),'FaceColor','interp','EdgeColor','none');
colormap(map)
daspect(gca,[1,1,4])
lightangle(305,30);
set(gcf,'Renderer','zbuffer'); lighting phong
isonormals(Ds hiso)
set(hcap,'AmbientStrength',6)
set(hiso,'SpecularColorReflectance',0,'SpecularExponent',50)
set(gca,'View',[215,30],'Box','On');
axis tight
title('Approximation at level 2')
displayEndOfDemoMessage(mfilename).

```

Figure 4.36 MATLAB Code

The tools developed by the Astronomical Medicine project will help pave the way for advanced 3D application of general scientific visualization and with continual development help improve the overall diagnostic experience. Three-Dimensional displaying of parts of the human body obtained by modern

diagnostic imaging methods will enable the displaying of Tomographic layers [27].

Observing layers of tissue on a pixel by pixel level would be a future goal of 3D reconstruction.

5.0 Conclusions

Chapter 5

From the experimental and mathematical results it can be concluded that for salt and pepper noise, the median filter is optimal compared to mean filter. It produces the maximum PSNR for the output image compared to the linear filters considered. From the output images shown in Chapter 3, the image obtained from the median filter has no noise present in it and is close to the high quality image. The sharpness of the image is retained unlike in the case of linear filtering. In the case where an image is corrupted with Gaussian noise, the wavelet shrinkage denoising has proved to be nearly optimal. The VisuShrink and the BayesShrink methods produce good PSNR values, often over 30 db. The Surelet method appears to be less effective when compared to VisuShrink and BayesShrink. The output from the BayesShrink method, slightly less effective than the VisuShrink did present a high quality image. It has been observed that BayesShrink is not effective for noise variance higher than 0.05. Denoising salt and pepper noise using BayesShrink was not effective. The denoised image in Image 3.27 (salt and pepper) shows very little improvement after application of the BayesShrink method.

The **Sym2** type and **Sym4** type of wavelet produce the best results when applied to MRI scans of the brain. Overall, the results of denoising the brain MRI images were disappointing with the maximum PSNR value recorded being 16.3 db. A value over 20 db would be a more desirable result concerning the image denoising criteria.

6.0 Scope for Future Work

Chapter 6

6.1 Fuzzy Methods

Enhancement of noisy image data is a challenging issue in many research and application areas. In the last few years, non-linear filters, feature extraction, and high dynamic range imaging methods based on soft computing models have been shown to be very effective in removing noise without destroying the useful information contained in the image data [28]. Enhancement of noisy images is not a trivial task. The filtering action must distinguish between unwanted noise (to be removed) and image details (to be preserved or possibly enhance). Soft computing and especially evolutionary and **fuzzy systems based methods** can effectively complete this task outperforming conventional methods [28]. Genetic algorithms and evolutionary methods may proved to be advantageous in image analysis, search analysis, and optimization while fuzzy reasoning is well suited to model uncertainty that typically occurs when both noise cancellation and detail preservation (enhancement) represent critical issues. The number of different approaches to evolutionary and fuzzy image processing has been progressively increasing with different areas of image processing introducing new **soft computing (fuzzy) methods** [28].

6.2 Vision 3-D Map

Image processing is enabling new technologies in the science of biomedical engineering. A Professor at the University of Southern California, Professor Gérard Medioni, is developing break-thru technology which could enable **the blind to "see"** [32]. He and biomedical engineering Professor James Weiland have created a pair of glasses and vest that provides a type of "seeing eye" for the blind. The glasses observe the environment and the vest gives directions, much like a seeing-eye dog sees and leads. Using two cameras, the glasses create a 3-D image of the surrounding environment. Though the blind person cannot see this image, the computer can, using it to analyze the setting. With this image, the computer can distinguish between obstacles and open space. Using this information, the computer sends signals to a vest through four sensors; much like the brain sends messages to the body. The vest's micromotor sensors -- similar to the ones that vibrate in cellphones -- indicate if the person needs to take a step towards the right and then sensors in the right hip vibrate. Once developed, this system could transform the lives of many, giving them independence and dignity. Eventually it will let users recognize obstacles and objects like lights, street signs, books on a shelf and hopefully, finally, people [32].

6.3 Model-Based Medical Image Computing

Medical image computing (MIC) is an interdisciplinary field at the intersection of computer science, data science, electrical engineering, physics,

mathematics and medicine [24]. This field develops computational and mathematical methods for solving problems pertaining to medical images and their use for biomedical research and clinical care [24]. The main goal of MIC is to extract clinically relevant information or knowledge from medical images. While closely related to the field of medical imaging, MIC focuses on the computational analysis of the images, not their acquisition. The methods can be grouped into several broad categories: image segmentation, image registration, image-based physiological modeling, and others. An example of image segmentation is shown in the image below [24].

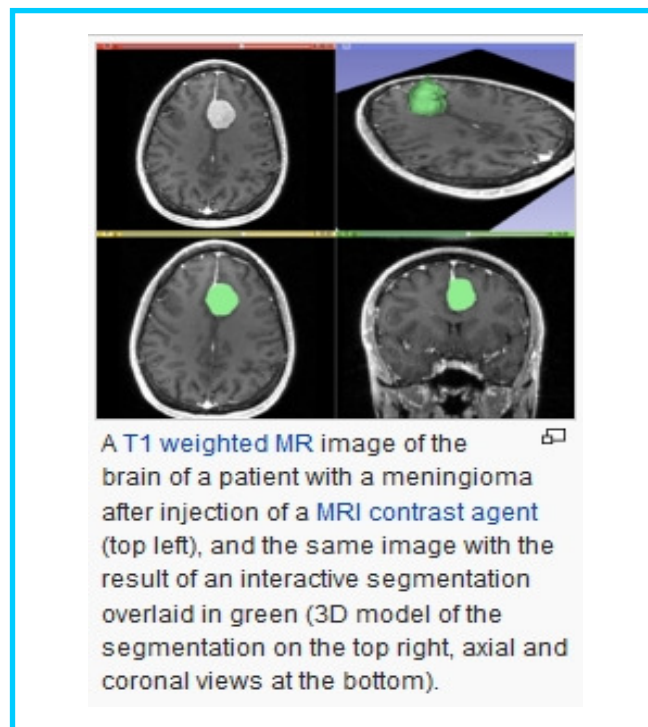


Figure 6.1 T1 Weighted MR Image

The segmentation of the 3D model is color coded in green. Medical image computing is especially important in medical diagnostics and image-guided therapy [45]. Image analysis systems integrating advanced image computing

methods can be used to extract quantitative image parameters or to support the surgeon during navigated intervention. The grade of automation, accuracy, reproducibility and robustness of medical image computing methods can be increased to meet the requirements in advanced clinical routines. Modeling approaches and model-based image analysis methods are new trends developed in model-based medical image computing [24]. Complex images like radiographic images, dual-energy CT images, MR images, diffusion tensor images as well as microscopic images are analyzed. Image-based models enable new insights and can lead to a deeper understanding of complex dynamic mechanisms in the human body. Model-based image computing methods will be important tools to improve the medical diagnostics and patient treatment of the future.

7.0 References

- [1] Anestis, Antoniadis and Jeremie, Bigot, 2001, Wavelet Estimators in Nonparametric Regression: A Comparative Simulation Study, *Journal of Statistical Software*, vol. 6, issue 6, pp. 1-83.
- [2] Baxes, Gregory, 1994, *Digital Image Processing*, John Wiley and Sons, New York, pp. 350-368.
- [3] Bhattacharya, Samayita, 2014, Comparative Study of Different Filters on Images in Frequency Domain, *International Journal of Advanced Research in Computer Science*, vol. 4, issue 8, pp. 948-954.
- [4] Borkin, Michelle, Goodman, Ayssa, Alan, Douglas, Kauffmann, Jens, and Halle, Michael, 2012, Application of Medical Imaging to the 3D Visualization of Astronomy Data, *Harvard University Initiative in Innovative Computing*, pp. 1-2.
- [5] Buades, Antoni, 2002, *Image and film denoising by non-local means*, Ph. D. Thesis, Universitat de les Illes Balears.
- [6] Chang, S. Grace, Bin, Yu and Martin, Vetterli, 2000, Adaptive Wavelet Thresholding for Image Denoising and Compression, *IEEE Trans, Image Processing*, vol. 9, issue 9, pp. 1532-1546.
- [7] Dangeti, Sarita, 2003, *Denoising Techniques - A Comparison*, Master Thesis, Louisiana State University.
- [8] Dewangan, Namrata, 2012, Adaptive Wavelet Thresholding for Image Denoising Using Various Shrinkage under Different Noise Conditions, *International Journal of Engineering Research and Technology*, vol. 1, issue 8, pp. 1-6.
- [9] Digital Image, 2016, 3D MRI Color Image, Citing Source: <http://www.deckeng.com>

-
- [10] Digital Image, 2016, 3D MRI Color Image,
Citing Source: <http://www.mayfieldclinic.com>
- [11] Digital Image, 2016, 3D MRI Color Image,
Citing Source: <http://www.fineartamerica.com>
- [12] Digital Image, 2016, 3D MRI Color Image,
Citing Source: <http://www.allposter.com>
- [13] Digital Image, 2016, 3D MRI Color Image,
Citing Source: <http://www.people.bath.ac.uk>.
- [14] Dixit, Arun, 2014, A Comparative Study of Wavelet Thresholding for Image Denoising, *I.J. Image Graphics and Signal Processing*, vol. 12, pp. 39-46.
- [15] Dogra, Ayush, 2013, Performance Comparision of Gaussian and Butterworth High Pass Filters, *Int. Journal of Innovations in Engineering*, vol. 2, no. 2, pp. 1-7.
- [16] Donoho, David L. and Iain M., Johnstone, 1994, Adapting to Unknown Smoothness with Wavelet Shrinkage, *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200-1224.
- [17] Edwards, Tim, 1992, Discrete Wavelet Transforms: Theory and Implementation, Discrete Wavelet Transforms, *Stanford University Press*, pp. 1-20.
- [18] Gonzalez, Rafael C., 2008, *Digital Image Processing*, 3rd Edition Prentice Hall, Upper Saddle River, pp. 313-393.
- [19] Gonzalez, Rafael C. and Woods, Richard E, 2004, *Digital Image Processing Using MATLAB*, Prentice Hall, Upper Saddle River, pp. 242-281.
- [20] Huynh-Thu, Q. and Ghanbari, M., 2008, Scope of validity of PSNR in Image/Video Quality Assessment, *Electronics Letters*, vol. 44, no. 13, pp. 800-801.
-

-
- [21] Ismail, B., 2012, Image De-noising with a New Threshold Value Using Wavelets, *Journal of Data Science*, vol. 10, pp. 259-270.
 - [22] Jain, Anil K., 1989, *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliffs, pp. 132-185.
 - [23] Kaur, Manpreet, 2013, A Survey on Implementation of Discrete Wavelet Transform for Image Processing, *Journal of Communication Networking*, vol. 2, issue 1, pp. 158-163.
 - [24] Karasev, P., 2015, Interactive MRI segmentation with Controlled Active Vision, *Proceedings of the Conference on Decision and Control*, pp. 2293-2298.
 - [25] Karu, Zoher, 2002, *Signals and Systems Made Ridiculously Simple*, ZiZi Press, Cambridge, pp. 77-100.
 - [26] Kim, Sung, 2013, *Application of Convolution in Image Processing with Matlab*, Univ. of Washington Press, Seattle, pp. 1-8.
 - [27] Kleut, Duska, Jovanovic, Milorad, and Reljin, Branimir, 2007, 3D Visualization of MRI images using MATLAB, *Journal of Automatic Control*, vol. 16, pp. 1-3.
 - [28] Krig, Scott, 2016, Future of Computer Vision, *Vision Systems Design*, vol. 21, no. 3, pp. 4-5.
 - [29] Leavline, E. JEB, 2013, A Comprehensive Study on Wavelet Based Shrinkage Methods for Denoising Natural Images, *Transactions on Signal Processing*, vol. 9, issue 4, pp. 203-215.
 - [30] Luisier, Florian, 2008, SURE-LET Multichannel Image Denoising: Interscale Orthonormal Wavelet Thresholding, *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 482-492.
 - [31] Matlab 6.1, 2005, Image Processing Toolbox,
Source: <http://www.mathworks.com/access/helpdesk/help/toolbox/images/images>
-

-
- [32] Medioni, Gerard, 2014, Visual Navigation aid for the blind in Dynamic Environments, *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 565-572.
- [33] Musoko, Victor, 2005, *Biomedical Signal and Image Processing*, Ph.D, Institute of Chemical Technology, Prague.
- [34] Mukesh, C. Motwani, 2001, *Survey of Image Denoising Techniques*, University of Nevada Press, Reno, pp. 1-20.
- [35] Nagu, Malothu, 2014, Image De-Noising By Using Median Filter and Wiener Filter, *Int. Journal of Innovative Research*, vol. 2, issue 9, pp. 5641-5649.
- [36] Nixon, Mark, 2009, *Feature Extraction & Image Processing*, Academic Press, Oxford, pp. 33-68.
- [37] Parker, J. R., 1997, *Image Algorithms for Image Processing and Computer Vision*, John Wiley and Sons, Incorporated, New York, pp. 220-249.
- [38] Raghuram, Rangarajan, 2002, *Image Denoising Using Wavelets*, University of Michigan, Ann Arbor, pp. 20-40.
- [39] Ruikar, Sachin D, 2011, Wavelet Based Image Denoising Technique, *Int. Journal of Advanced Computer Science*, vol. 2, no. 3, pp. 49-53.
- [40] Siddeq, Mohammed M., 2009, Using Discrete Wavelet Transform and Wiener filter Image Denoising, *Wasit Journal for Science & Medicine*, vol. 2, no. 2, pp. 18-30.
- [41] Sidhu, Kanwaljot Singh, 2012, Medical Image Denoising In the Wavelet Domain Using Haar and DB3 Filtering, *International Refereed Journal of Engineering and Science*, vol. 1, issue 1, pp. 1-8.
- [42] Smith, W. Steven, 1999, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, San Diego, pp. 172-209.
-

-
- [43] Sonali, Mahakale R., 2012, A Comparative Study of Image Filtering on Various Noisy Pixels, *International Journal of Image Processing and Vision Sciences*, vol. 1, issue 2, pp. 69-77.
- [44] Suraj, A. Anoop, 2014, Discrete wavelet transform based image fusion and denoising in FPGA, *Journal of Electrical Systems and Information Technology*, vol. 1, pp. 72-81.
- [45] Yaroslavsky, L., 2004, *Digital Image Processing: Applications, Lecture Notes*, Department of Interdisciplinary Studies, Tel Aviv University, pp. 1-32.
- [46] Wen, Che-Yen, 2001, Point Spread Function and their Applications to Forensic Image Restoration, *Optical Engineering*, vol. 29, issue 5, pp. 1-12.